

Informe de Laboratorio 13

Tema: ClaseSoldado-MiembrosdeClase

Nota

Estudiante	Escuela	Asignatura
Sarayasi Huanaco,Jeferson Jesus	Escuela Profesional de Ingeniería de Sistemas	Fundamentos de la Programación 2

Laboratorio	Tema	Duración
13	ClaseSoldado-MiembrosdeClase	48 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - B		

Docente
Aedo Lopez, Marco Wilfredo

1. Tarea

2. Equipos, Materiales y temas utilizados

- Sistema Operativo Windows 11.
- VIM 9.0.
- OpenJHK 64-Bits 17.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional
- Arreglos Estandar

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar
- <https://github.com/JefersonSH/FP2-2024B.git>
- URL para el laboratorio 07 en el Repositorio GitHub.

4. Ejercicios Resuletos

Ejercicio 01

- **Commit 1:** VideoJuego numero 10 con las modificaciones pedidas por el docente

Listing 1: Clase Direccion

```
1 import java.util.*;
2
3 public class VideoJuego10 {
4     private static int tamaoMax = 10;
5
6     public static void Jugar(){
7         Scanner sc = new Scanner(System.in);
8         int turno = 1;
9
10        int tamao1 = (int)(Math.random()* tamaoMax + 1); // Soldados del ejercito 1
11        int tamao2 = (int)(Math.random()* tamaoMax + 1); // Soldados del ejercito 2
12        Soldado[] [] tablero = new Soldado[tamaoMax][tamaoMax]; // Creando un arreglo
13                               bidimensional de Soldados
14
15        AsignarAtributos(tablero, tamao1, 1); // Creando Soldados en el ejercito 0
16        AsignarAtributos(tablero, tamao2, 2); // Creando Soldados en el ejercito 1
17
18        ImprimirDatos(tablero);
19        ImprimirTablero(tablero);
20
21        System.out.println("Soldado con mas vida del ejercito 1 es: ");
22        EncontrarMayorVida(tablero, 1);
23        System.out.println("Soldado con mas vida del ejercito 2 es: ");
24        EncontrarMayorVida(tablero, 2);
25
26        System.out.println("Promedio de Vida del Ejercito 1: " + PromedioVida(tablero, 1) +
27                               "\n");
28        System.out.println("Promedio de Vida del Ejercito 2: " + PromedioVida(tablero, 2) +
29                               "\n");
30        System.out.println("\n-----\n");
31
32        System.out.println("Vida de todos los soldados:");
33        ImprimirVida(tablero);
34
35        System.out.println("Ranking de Vida segun Ordenamiento Burbuja");
36        OrdenamientoBurbuja(tablero, SoldadosVivos(tablero));
37
38        System.out.println("Ranking de Vida segun Ordenamiento por Insercion");
39        OrdenamientoInsercion(tablero, SoldadosVivos(tablero));
40
41        Jugada(tablero, turno);
42    }
43
44    public static void AsignarAtributos(Soldado[] [] tablero, int n, int ejercito){
45        for(int i = 0; i < n; i++){
46            int x = (int)(Math.random()*tamaoMax); // x representa Filas
```

```
46     int y = (int)(Math.random()*tamaoMax); // y representa a las columnas, siendo del 1
47     al 9
48     if(tablero[x][y] == null){ // Si la posicion [y][x] esta vacia, crea un Soldado
49         String nombre = "Soldado" + (ejercito) + "x" + (i + 1);
50         int vida = (int)(Math.random()* 5 + 1);
51         int ataque = (int)(Math.random()* 5 + 1);
52         int defensa = (int)(Math.random()* 5 + 1);
53         tablero[x][y] = new Soldado(nombre, vida, ejercito, x + 1, y + 1, ataque, defensa);
54     } else {
55         i--; // Caso contrario, se regresa al ciclo
56     }
57 }
58
59 public static void ImprimirDatos(Soldado[][] tablero){
60     for(int i = 0; i < tamaoMax; i++) // Filas
61         for(int j = 0; j < tamaoMax; j++) // Controla las columnas
62             if(tablero[i][j] != null)
63                 if(tablero[i][j].getVive()){
64                     System.out.print(tablero[i][j].getNombre() + " ");
65                     System.out.print("Salud: " + tablero[i][j].getVidaActual() + " ");
66                     System.out.println("Ejercito: " + tablero[i][j].getEjercito());
67                     System.out.print("Fila: " + tablero[i][j].getFila() + " ");
68                     System.out.println("Columna: " + tablero[i][j].getColumna());
69                     System.out.print("ATQ: " + tablero[i][j].getAtaque() + " ");
70                     System.out.println("DEF: " + tablero[i][j].getDefensa() + "\n");
71                 }
72     System.out.println("\n-----\n");
73 }
74
75 public static void ImprimirTablero(Soldado[][] tablero){
76     for(int i = 0; i < tamaoMax; i++){ // Filas
77         for(int j = 0; j < tamaoMax; j++){ // Columnas
78             if(tablero[i][j] != null) // Si el indice contiene un Soldado continua
79                 if(tablero[i][j].getVive()){ // Si el soldado esta vivo continua
80                     if(tablero[i][j].getEjercito() == 1) System.out.print("1 ");
81                     if(tablero[i][j].getEjercito() == 2) System.out.print("2 ");
82                 } else System.out.print("=" );
83             else System.out.print("=" );
84
85         }
86         System.out.print("\n"); // Una vez acabada una fila, hace un salto de linea
87     }
88     System.out.println("\n-----\n");
89 }
90
91 public static void ImprimirTablero(Soldado[][] tablero, int ejercito){
92     for(int i = 0; i < tamaoMax; i++){ // Filas
93         for(int j = 0; j < tamaoMax; j++){ // Columnas
94             if(tablero[i][j] != null) // Si el indice contiene un Soldado continua
95                 if(tablero[i][j].getVive()) // Si el soldado esta vivo continua
96                     if(tablero[i][j].getEjercito() == ejercito) System.out.print(ejercito + "
97 ");
98                     else System.out.print("=" );
99                     else System.out.print("=" );
100                     else System.out.print("=" );
```

```
100     }
101     System.out.print("\n"); // Una vez acabada una fila, hace un salto de linea
102 }
103 System.out.println("\n-----\n");
104 }
105
106 public static void EncontrarMayorVida(Soldado[][] tablero, int ejercito){
107     int mayor = Integer.MIN_VALUE;
108
109     Soldado masVida = new Soldado();
110     for(int i = 0; i < tamaoMax; i++) // Filas
111         for(int j = 0; j < tamaoMax; j++) // Columnas
112             if(tablero[j][i] != null)
113                 if(tablero[j][i].getVive()){
114                     if(tablero[j][i].getEjercito() == ejercito) // Pasa si son del mismo ejercito
115                         if(tablero[j][i].getVidaActual() > mayor){
116                             mayor = tablero[j][i].getVidaActual(); //Si es mayor, actualiza la
117                                 variable
118                             masVida = tablero[j][i];
119                         }
120                 }
121     Imprimir(masVida);
122 }
123
124 public static void Imprimir(Soldado soldado){
125     System.out.print(soldado.getNombre() + " ");
126     System.out.print("Salud: " + soldado.getVidaActual() + " ");
127     System.out.println("Ejercito: " + soldado.getEjercito());
128     System.out.print("Fila: " + soldado.getFila() + " ");
129     System.out.println("Columna: " + soldado.getColumna());
130     System.out.print("ATQ: " + soldado.getAtaque() + " ");
131     System.out.println("DEF: " + soldado.getDefensa() + "\n");
132 }
133
134 public static int PromedioVida(Soldado[][] tablero, int ejercito){
135     int promedio = 0;
136     int size = 0;
137     for(int i = 0; i < tamaoMax; i++) // Filas
138         for(int j = 0; j < tamaoMax; j++) // Columnas
139             if(tablero[j][i] != null)
140                 if(tablero[j][i].getVive() && tablero[j][i].getEjercito() == ejercito){
141                     promedio += tablero[j][i].getVidaActual();
142                     size++;
143                 }
144     return promedio/size;
145 }
146
147 public static void ImprimirVida(Soldado[][] tablero){
148     for(int i = 0; i < tamaoMax; i++) // Filas
149         for(int j = 0; j < tamaoMax; j++) // Columnas
150             if(tablero[j][i] != null)
151                 if(tablero[j][i].getVive())
152                     System.out.println("Vida de " + tablero[j][i].getNombre() + ": " +
153                         tablero[j][i].getVidaActual());
154     System.out.println("\n-----\n");
155 }
```

```
154
155 public static void OrdenamientoBurbuja(Soldado[] [] tablero, int n){
156     Soldado temp = new Soldado();
157     int contador = 0;
158     Soldado[] lista = new Soldado[n];
159     for(int i = 0; i < tamaoMax; i++) // Filas
160         for(int j = 0; j < tamaoMax; j++) // Columnas
161             if(tablero[j][i] != null){
162                 if(tablero[j][i].getVive()){
163                     lista[contador] = tablero[j][i];
164                     contador++;
165                 }
166             }
167
168     for(int i = 0; i < n - 1; i++){
169         for(int j = 0; j < n - 1; j++){
170             if(lista[j].getVidaActual() > lista[j+1].getVidaActual()){
171                 temp = lista[j+1];
172                 lista[j+1] = lista[j];
173                 lista[j] = temp;
174             }
175         }
176         for(int i = n - 1; i >= 0; i--){
177             Imprimir(lista[i]);
178         }
179         System.out.println("\n-----\n");
180     }
181
182     public static void OrdenamientoInsercion(Soldado[] [] tablero, int n){
183         Soldado temp = new Soldado();
184         int contador = 0;
185         Soldado[] lista = new Soldado[n];
186         for(int i = 0; i < tamaoMax; i++) // Filas
187             for(int j = 0; j < tamaoMax; j++) // Columnas
188                 if(tablero[j][i] != null){
189                     if(tablero[j][i].getVive()){
190                         lista[contador] = tablero[j][i];
191                         contador++;
192                     }
193                 }
194
195         for(int i = 1; i < n; i++){
196             Soldado key = lista[i];
197             int j = i - 1;
198             while(j >= 0 && lista[j].getVidaActual() > key.getVidaActual()){
199                 lista[j + 1] = lista[j];
200                 j = j - 1;
201             }
202             lista[j + 1] = key;
203         }
204
205         for(int i = n - 1; i >= 0; i--){
206             Imprimir(lista[i]);
207         }
208         System.out.println("\n-----\n");
209     }
```

```
210 public static void Jugada(Soldado[][] tablero, int turno){
211     Scanner sc = new Scanner(System.in);
212     boolean condicion = true;
213     System.out.println("\nSe han creado " + SoldadosVivos(tablero) + " Soldados\n");
214
215     while(condicion){
216
217         turno %= 2; //Los turnos pares seran los turnos del jugador 2
218         if(turno == 0) turno = 2; //Si es turno impar, jugara el jugador 1
219         //Se pide al usuario la jugada y se verifica si es valida
220         System.out.println("Si quieres salir al menu principal, coloca *cancelar* en
                cualquier momento");
221         System.out.println("Jugador " + turno + ". Ingrese el soldado a mover del Ejercito "
                + turno + ". (Ejm. Soldado1x1)");
222         String soldado = sc.nextLine();
223         boolean condicion1 = VerificarSoldado(soldado, turno, tablero);
224
225         if(soldado.equals("cancelar")){
226             return;
227         }
228
229         while(!condicion1){
230             System.out.println("Soldado no encontrado, por favor ingrese otro");
231             soldado = sc.nextLine();
232             if(soldado.equals("cancelar")){
233                 return;
234             }
235             condicion1 = VerificarSoldado(soldado, turno, tablero);
236         }
237
238         System.out.println("Ahora ingrese la direccion (w:up, s:down, d:right, a:left)");
239         String direccion = sc.nextLine();
240
241         if(direccion.equals("cancelar")){
242             return;
243         }
244
245         boolean condicion2 = VerificarDireccion(direccion, turno, tablero, soldado);
246         while(!condicion2){
247             System.out.println("Direccion no admitida, por favor ingrese otro");
248             direccion = sc.nextLine();
249             if(direccion.equals("cancelar")){
250                 return;
251             }
252             condicion2 = VerificarDireccion(direccion, turno, tablero, soldado);
253         }
254         //Si hay enfrentamiento, se realiza
255         SoldadoMovriendose(tablero, turno, soldado, direccion);
256
257         ImprimirTablero(tablero); //Despues de la jugada, se imprime el tablero y los datos
                de los soldados vivos
258         ImprimirDatos(tablero);
259         System.out.println("\nHay " + SoldadosVivos(tablero, 1) + " Soldados vivos del
                ejercito 1\n");
260         System.out.println("\nHay " + SoldadosVivos(tablero, 2) + " Soldados vivos del
                ejercito 2\n");
```

```
261         condicion = ActualizarCondicion(tablero); // Si no hay soldados vivos en ambos
262             ejercitos, sera false
263
264         turno++;
265     }
266
267     Ganador(tablero);
268
269 }
270
271
272 public static int SoldadosVivos(Soldado[] [] tablero){
273     int contador = 0;
274     for(int i = 0; i < tamaoMax; i++) // Filas
275         for(int j = 0; j < tamaoMax; j++) // Columnas
276             if(tablero[i][j] != null)
277                 if(tablero[i][j].getVive())
278                     contador++;
279     return contador;
280 }
281
282 public static int SoldadosVivos(Soldado[] [] tablero, int ejercito){
283     int contador = 0;
284     for(int i = 0; i < tamaoMax; i++) //Controla las filas
285         for(int j = 0; j < tamaoMax; j++) //Controla las columnas
286             if(tablero[i][j] != null)
287                 if(tablero[i][j].getVive() && tablero[i][j].getEjercito() == ejercito)
288                     contador++;
289     return contador;
290 }
291
292 public static boolean VerificarSoldado(String soldado, int turno, Soldado[] [] tablero){
293     for(int i = 0; i < tamaoMax; i++) // Filas
294         for(int j = 0; j < tamaoMax; j++) // Columnas
295             if(tablero[i][j] != null)
296                 if(tablero[i][j].getVive() && tablero[i][j].getEjercito() == turno)
297                     if(tablero[i][j].getNombre().equals(soldado))
298                         return true;
299     return false;
300 }
301
302 public static boolean VerificarDireccion(String direccion, int turno, Soldado[] []
303     tablero, String soldado){
304     if(!(direccion.equals("w") || direccion.equals("s") || direccion.equals("a") ||
305         direccion.equals("d")))
306         return false;
307
308     for(int i = 0; i < tamaoMax; i++) { // Filas
309         for(int j = 0; j < tamaoMax; j++) { // Columnas
310             if(tablero[i][j] != null && tablero[i][j].getVive() && tablero[i][j].getEjercito()
311                 == turno && tablero[i][j].getNombre().equals(soldado)){
312                 if(direccion.equals("w") && i - 1 >= 0) // Verifica movimiento hacia arriba
313                     return true;
314                 else if(direccion.equals("s") && i + 1 < tamaoMax) // Verifica movimiento hacia
315                     abajo
```

```
312         return true;
313     else if(direccion.equals("a") && j - 1 >= 0) // Verifica movimiento hacia la
        izquierda
314         return true;
315     else if(direccion.equals("d") && j + 1 < tamaoMax) // Verifica movimiento hacia
        la derecha
316         return true;
317     }
318 }
319 }
320 return false; //Si no devuelve true, el movimiento no es valido
321 }
322
323 public static void SoldadoMovriendose(Soldado[][] tablero, int turno, String soldado,
    String direccion){
324     int fila, columna;
325     int contador = 0; // Este contador es MUY IMPORTANTE, evitara que se llame el metodo
        Combate() mas de 1 vez, evitando posibles errores con s o d
326     for(int i = 0; i < tamaoMax; i++) // Filas
327         for(int j = 0; j < tamaoMax; j++) // Columnas
328             if(tablero[i][j] != null && tablero[i][j].getVive() && tablero[i][j].getEjercito()
                == turno && tablero[i][j].getNombre().equals(soldado) && contador == 0){
329                 if(direccion.equals("w") && i - 1 >= 0){ // Verifica movimiento hacia arriba
330                     Combate(tablero, i, j, direccion);
331                     contador++;
332                 }else if(direccion.equals("s") && i + 1 < tamaoMax){ // Verifica movimiento
333                     hacia abajo
334                     Combate(tablero, i, j, direccion);
335                     contador++;
336                 }else if(direccion.equals("a") && j - 1 >= 0){ // Verifica movimiento hacia la
337                     izquierda
338                     Combate(tablero, i, j, direccion);
339                     contador++;
340                 }else if(direccion.equals("d") && j + 1 < tamaoMax){ // Verifica movimiento
341                     hacia la derecha
342                     Combate(tablero, i, j, direccion);
343                     contador++;
344                 }
345             }
346         }
347     }
348
349     public static void Combate(Soldado[][] tablero, int x, int y, String direccion){
350         if(direccion.equals("w")){ //La estructura se repite para cada una de las 4 letras,
            pero con sus variaciones
351             if(tablero[x - 1][y] == null || !tablero[x - 1][y].getVive()){ //Verifica que no
                haya un Soldado vivo en el casillero a avanzar
352                 tablero[x - 1][y] = new Soldado(tablero[x][y].getNombre(),
                    tablero[x][y].getVidaActual(), tablero[x][y].getEjercito(), x + 1 - 1, y + 1);
353                 tablero[x][y].morir();
354             } else {
355                 int num = (int)(Math.random() * 100 + 1);
356                 double total = tablero[x][y].getVidaActual() + tablero[x - 1][y].getVidaActual();
357                 double probabilidadA = 100 * (tablero[x][y].getVidaActual() / total);
358                 System.out.println("Probabilidades de ganar:\nSoldado actual: " + probabilidadA +
                    "% --- Soldado enemigo: " + (100 - probabilidadA) + "%");
359                 if(num < probabilidadA){ // El ganador seria el soldado 1
```



```
356         tablero[x - 1][y] = new Soldado(tablero[x][y].getNombre(),
357             tablero[x][y].getVidaActual() + 1, tablero[x][y].getEjercito(), x + 1 - 1,
358             y + 1);
359         tablero[x][y].morir();
360         System.out.println("El Soldado actual gana");
361     } else { // Si no el ganador seria el soldado 2
362         tablero[x - 1][y].setVidaActual(tablero[x - 1][y].getVidaActual() + 1);
363         tablero[x][y].morir();
364         System.out.println("El Soldado enemigo gana");
365     }
366 }
367 } else if(direccion.equals("s")){
368     if(tablero[x + 1][y] == null || !tablero[x + 1][y].getVive()){
369         tablero[x + 1][y] = new Soldado(tablero[x][y].getNombre(),
370             tablero[x][y].getVidaActual(), tablero[x][y].getEjercito(), x + 1 + 1, y + 1);
371         tablero[x][y].morir();
372     } else {
373         int num = (int)(Math.random() * 100 + 1);
374         double total = tablero[x][y].getVidaActual() + tablero[x + 1][y].getVidaActual();
375         double probabilidadA = 100 * (tablero[x][y].getVidaActual() / total);
376         System.out.println("Probabilidades de ganar:\nSoldado actual: " + probabilidadA +
377             "% --- Soldado enemigo: " + (100 - probabilidadA) + "%");
378         if(num < probabilidadA){ // El ganador seria el soldado 1
379             tablero[x + 1][y] = new Soldado(tablero[x][y].getNombre(),
380                 tablero[x][y].getVidaActual() + 1, tablero[x][y].getEjercito(), x + 1 - 1,
381                 y + 1);
382             tablero[x][y].morir();
383             System.out.println("El Soldado actual gana");
384         } else { // Si no el ganador seria el soldado 2
385             tablero[x + 1][y].setVidaActual(tablero[x + 1][y].getVidaActual() + 1);
386             tablero[x][y].morir();
387             System.out.println("El Soldado enemigo gana");
388         }
389     }
390 }
391 } else if(direccion.equals("a")){
392     if(tablero[x][y - 1] == null || !tablero[x][y - 1].getVive()){
393         tablero[x][y - 1] = new Soldado(tablero[x][y].getNombre(),
394             tablero[x][y].getVidaActual(), tablero[x][y].getEjercito(), x + 1, y + 1 - 1
395             );
396         tablero[x][y].morir();
397     } else {
398         int num = (int)(Math.random() * 100 + 1);
399         double total = tablero[x][y].getVidaActual() + tablero[x][y - 1].getVidaActual();
400         double probabilidadA = 100 * (tablero[x][y].getVidaActual() / total);
401         System.out.println("Probabilidades de ganar:\nSoldado actual: " + probabilidadA +
402             "% --- Soldado enemigo: " + (100 - probabilidadA) + "%");
403         if(num < probabilidadA){ // El ganador seria el soldado 1
404             tablero[x][y - 1] = new Soldado(tablero[x][y].getNombre(),
405                 tablero[x][y].getVidaActual() + 1, tablero[x][y].getEjercito(), x + 1 - 1,
406                 y + 1);
407             tablero[x][y].morir();
408             System.out.println("El Soldado actual gana");
409         } else { // Si no el ganador seria el soldado 2
410             tablero[x][y - 1].setVidaActual(tablero[x][y - 1].getVidaActual() + 1);
411             tablero[x][y].morir();
412             System.out.println("El Soldado enemigo gana");
413         }
414     }
415 }
```

```
401     }
402   }
403   } else if(direccion.equals("d")){
404     if(tablero[x][y + 1] == null || !tablero[x][y + 1].getVive()){
405       tablero[x][y + 1] = new Soldado(tablero[x][y].getNombre(),
406         tablero[x][y].getVidaActual(), tablero[x][y].getEjercito(), x + 1, y + 1 + 1);
407       tablero[x][y].morir();
408     } else {
409       int num = (int)(Math.random() * 100 + 1);
410       double total = tablero[x][y].getVidaActual() + tablero[x][y + 1].getVidaActual();
411       double probabilidadA = 100 * (tablero[x][y].getVidaActual() / total);
412       System.out.println("Probabilidades de ganar:\nSoldado actual: " + probabilidadA +
413         "% --- Soldado enemigo: " + (100 - probabilidadA) + "%");
414       if(num < probabilidadA){ // El ganador seria el soldado 1
415         tablero[x][y + 1] = new Soldado(tablero[x][y].getNombre(),
416           tablero[x][y].getVidaActual() + 1, tablero[x][y].getEjercito(), x + 1 - 1,
417             y + 1);
418         tablero[x][y].morir();
419         System.out.println("El Soldado actual gana");
420       } else { // Si no el ganador seria el soldado 2
421         tablero[x][y + 1].setVidaActual(tablero[x][y + 1].getVidaActual() + 1);
422         tablero[x][y].morir();
423         System.out.println("El Soldado enemigo gana");
424       }
425     }
426   }
427 }
428
429 public static boolean ActualizarCondicion(Soldado[][] tablero){
430   int ejercito1 = SoldadosVivos(tablero, 1);
431   int ejercito2 = SoldadosVivos(tablero, 2);
432   if(ejercito1 == 0 || ejercito2 == 0)
433     return false;
434   return true;
435 }
436
437 public static void Ganador(Soldado[][] tablero){
438   if(SoldadosVivos(tablero, 1) == 0)
439     System.out.println("El Ganador es el Ejercito 2");
440   else if(SoldadosVivos(tablero, 2) == 0)
441     System.out.println("El Ganador es el Ejercito 1");
442   else System.out.println("Error");
443 }
444
445 public static boolean CrearSoldado(String texto, Soldado[][] tablero, int
446   ejercitoOriginal){
447   //si retorna false, nos saca del bucle, pero si retorna true, el bucle sigue
448   if(texto.length() - texto.replace(",","").length() != 6){
449     System.out.println("Datos NO VALIDOS, por favor ingrese nuevos datos");
450     return true;
451   }
452
453   String nombre = texto.substring(0,texto.indexOf(","));
454   texto = texto.substring(texto.indexOf(",") + 1,texto.length());
455   int vida = Integer.parseInt(texto.substring(0,texto.indexOf(",")));
456   texto = texto.substring(texto.indexOf(",") + 1,texto.length());
```

```
452     int ejercito = Integer.parseInt(texto.substring(0,texto.indexOf(",")));
453     texto = texto.substring(texto.indexOf(",") + 1,texto.length());
454     int fila = Integer.parseInt(texto.substring(0,texto.indexOf(",")));
455     texto = texto.substring(texto.indexOf(",") + 1,texto.length());
456     int columna = Integer.parseInt(texto.substring(0,texto.indexOf(",")));
457     texto = texto.substring(texto.indexOf(",") + 1,texto.length());
458     int ataque = Integer.parseInt(texto.substring(0,texto.indexOf(",")));
459     texto = texto.substring(texto.indexOf(",") + 1,texto.length());
460     int defensa = Integer.parseInt(texto);
461     if(ejercito != ejercitoOriginal){
462         System.out.println("Los EJERCITOS NO COINCIDEN, por favor ingrese nuevos datos");
463         return true;
464     }
465     if(tablero[fila - 1][columna - 1] == null){
466         tablero[fila - 1][columna - 1] = new Soldado(nombre, vida, ejercito, fila, columna,
467             ataque, defensa);
468         Imprimir(tablero[fila - 1][columna - 1]);
469         ImprimirTablero(tablero);
470         return false;
471     }else if(!tablero[fila - 1][columna - 1].getVive()){
472         tablero[fila - 1][columna - 1] = new Soldado(nombre, vida, ejercito, fila, columna,
473             ataque, defensa);
474         Imprimir(tablero[fila - 1][columna - 1]);
475         ImprimirTablero(tablero);
476         return false;
477     }else System.out.println("Posicion NO VALIDA, por favor ingrese nuevos datos");
478     return true;
479 }
480
481 public static boolean EliminarSoldado(String texto, Soldado[][] tablero, int ejercito){
482     //si retorna false, nos saca del bucle, pero si retorna true, el bucle sigue
483     for(int i = 0; i < tamaoMax; i++)
484         for(int j = 0; j < tamaoMax; j++)
485             if(tablero[i][j] != null)
486                 if(tablero[i][j].getNombre().equals(texto) && tablero[i][j].getVive()){
487                     if(ejercito != tablero[i][j].getEjercito()){
488                         System.out.println("Los EJERCITOS NO COINCIDEN, por favor ingrese nuevos
489                             datos");
490                         return true;
491                     }
492                     tablero[i][j].morir();
493                     ImprimirTablero(tablero);
494                     return false;
495                 }
496     }
497     System.out.println("Soldado no encontrado, por favor ingrese un nombre valido");
498     return true;
499 }
500
501 public static boolean ClonarSoldado(String texto, Soldado[][] tablero, int ejercito){
502     //si retorna false, nos saca del bucle, pero si retorna true, el bucle sigue
503     for(int i = 0; i < tamaoMax; i++)
504         for(int j = 0; j < tamaoMax; j++)
505             if(tablero[i][j] != null)
506                 if(tablero[i][j].getNombre().equals(texto) && tablero[i][j].getVive()){
507                     if(ejercito != tablero[i][j].getEjercito()){
508                         System.out.println("Los EJERCITOS NO COINCIDEN, por favor ingrese nuevos
```

```
505         datos");
506         return true;
507     }
508     int x = tamaoMax - 1, y = tamaoMax - 1;
509     while(x >= 0){
510         if(tablero[x][y] == null || !tablero[x][y].getVive()){
511             tablero[x][y] = new Soldado(tablero[i][j].getNombre(),
512                 tablero[i][j].getVidaActual(), ejercito, x + 1, y + 1,
513                 tablero[i][j].getAtaque(), tablero[i][j].getDefensa());
514             ImprimirTablero(tablero);
515             return false;
516         } else {
517             if(y > 0) y--;
518             else{
519                 x--;
520                 y = tamaoMax - 1;
521             }
522         }
523     }
524 }
525
526 System.out.println("Soldado no encontrado, por favor ingrese un nombre valido");
527 return true;
528 }
529
530 public static boolean ModificarSoldado(String texto, Soldado[][] tablero, int ejercito){
531     //si retorna false, nos saca del bucle, pero si retorna true, el bucle sigue
532     Scanner sc = new Scanner(System.in);
533     for(int i = 0; i < tamaoMax; i++)
534         for(int j = 0; j < tamaoMax; j++)
535             if(tablero[i][j] != null)
536                 if(tablero[i][j].getNombre().equals(texto) && tablero[i][j].getVive()){
537                     boolean condicion1 = true;
538                     while(condicion1){ // El bucle termina cuando se ingrese un numero valido
539                         System.out.println("Seleccione que quiere modificar:\n1. VIDA\n2.
540                             ATAQUE\n3. DEFENSA\n4. SALIR\n");
541                         int opcion = sc.nextInt();
542                         boolean condicion2 = true;
543                         if(opcion == 1){
544                             while(condicion2){ // EL bucle termina cuando se ingrese un valor
545                                 valido, mayor a 0 y menor a 6
546                                 System.out.println("Ingrese el nuevo valor de VIDA");
547                                 int vida = sc.nextInt();
548                                 tablero[i][j].setVidaActual(vida);
549                                 if(0 < vida && vida < 6) condicion2 = false;
550                             }
551                         } else if(opcion == 2){
552                             while(condicion2){ // EL bucle termina cuando se ingrese un valor
553                                 valido, mayor a 0 y menor a 6
554                                 System.out.println("Ingrese el nuevo valor de ATAQUE");
555                                 int ataque = sc.nextInt();
556                                 tablero[i][j].setAtaque(ataque);
557                                 if(0 < ataque && ataque < 6) condicion2 = false;
558                             }
559                         } else if(opcion == 3){
560                             while(condicion2){ // EL bucle termina cuando se ingrese un valor
561                                 valido, mayor a 0 y menor a 6
```

```
554         System.out.println("Ingrese el nuevo valor de DEFENSA");
555         int defensa = sc.nextInt();
556         tablero[i][j].setDefensa(defensa);
557         if(0 < defensa && defensa < 6) condicion2 = false;
558     }
559     } else if(opcion == 4){
560         condicion1 = false;
561     } else {
562         System.out.println("Valor no valido. Ingrese de nuevo una opcion
563                             valida");
564     }
565     return false;
566 }
567 System.out.println("Soldado no encontrado, por favor ingrese un nombre valido");
568 return true;
569 }
570
571 public static boolean CompararSoldados(String[] nombres, Soldado[][] tablero, int
572     ejercito){
573     //si retorna false, nos saca del bucle, pero si retorna true, el bucle sigue
574     Scanner sc = new Scanner(System.in);
575     int contador = 0;
576     int[][] datos = new int[2][3];
577     String[] soldados = new String[2];
578     for(int i = 0; i < tamaoMax; i++)
579         for(int j = 0; j < tamaoMax; j++)
580             if(tablero[i][j] != null)
581                 if(contador < nombres.length && (tablero[i][j].getNombre().equals(nombres[0])
582                     || tablero[i][j].getNombre().equals(nombres[1])) &&
583                     tablero[i][j].getVive()){
584                     if(ejercito != tablero[i][j].getEjercito()){
585                         System.out.println("Los EJERCITOS NO COINCIDEN, por favor ingrese nuevos
586                                             datos");
587                         return true;
588                     }
589                     if(contador < 2){
590                         soldados[contador] = tablero[i][j].getNombre();
591                         datos[contador][0] = tablero[i][j].getVidaActual();
592                         datos[contador][1] = tablero[i][j].getAtaque();
593                         datos[contador][2] = tablero[i][j].getDefensa();
594                         contador++;
595                     }
596                 }
597             }
598     if(contador < 2){
599         System.out.println("Soldados no encontrados. Por favor ingrese nombres validos");
600         return true;
601     }
602     String[] condiciones = {"NO", "NO", "NO", "NO"};
603     if(soldados[0].equals(soldados[1])) condiciones[0] = "SI";
604     if(datos[0][0] == datos[1][0]) condiciones[1] = "SI";
605     if(datos[0][1] == datos[1][1]) condiciones[2] = "SI";
606     if(datos[0][2] == datos[1][2]) condiciones[3] = "SI";
607     System.out.println("Los nombres " + condiciones[0] + " son iguales\n" +
608         "Los atributos de VIDA " + condiciones[1] + " son iguales\n" +
609         "Los atributos de ATAQUE " + condiciones[2] + " son iguales\n" +
```

```
605     "Los atributos de DEFENSA " + condiciones[3] + " son iguales\n");
606     return false;
607 }
608
609 public static boolean IntercambiarSoldados(String[] nombres, Soldado[][] tablero, int
    ejercito){
610     //si retorna false, nos saca del bucle, pero si retorna true, el bucle sigue
611     Scanner sc = new Scanner(System.in);
612     int contador = 0;
613     int[][] datos = new int[2][2];
614     for(int i = 0; i < tamaoMax; i++)
615         for(int j = 0; j < tamaoMax; j++)
616             if(tablero[i][j] != null)
617                 if(contador < nombres.length && (tablero[i][j].getNombre().equals(nombres[0])
                    || tablero[i][j].getNombre().equals(nombres[1]) ) &&
                    tablero[i][j].getVive()){
618                     if(ejercito != tablero[i][j].getEjercito()){
619                         System.out.println("Los EJERCITOS NO COINCIDEN, por favor ingrese nuevos
                            datos");
620                         return true;
621                     }
622                     if(contador < 2){
623                         datos[contador][0] = i;
624                         datos[contador][1] = j;
625                         contador++;
626                     }
627                 }
628     if(contador < 2){
629         System.out.println("Soldados no encontrados. Por favor ingrese nombres validos");
630         return true;
631     }
632     Soldado soldado1 = new Soldado(tablero[datos[0][0]][datos[0][1]].getNombre(),
        tablero[datos[0][0]][datos[0][1]].getVidaActual(), ejercito, datos[1][0] + 1,
        datos[1][1] + 1, tablero[datos[0][0]][datos[0][1]].getAtaque(),
        tablero[datos[0][0]][datos[0][1]].getDefensa());
633     Soldado soldado2 = new Soldado(tablero[datos[1][0]][datos[1][1]].getNombre(),
        tablero[datos[1][0]][datos[1][1]].getVidaActual(), ejercito, datos[0][0] + 1,
        datos[0][1] + 1, tablero[datos[1][0]][datos[1][1]].getAtaque(),
        tablero[datos[1][0]][datos[1][1]].getDefensa());
634     tablero[datos[0][0]][datos[0][1]] = soldado2;
635     tablero[datos[1][0]][datos[1][1]] = soldado1;
636     System.out.println("Los soldados se han intercambiado de posicion\n");
637     ImprimirTablero(tablero);
638     return false;
639 }
640
641 public static boolean ImprimirSoldado(String texto, Soldado[][] tablero, int ejercito){
642     //si retorna false, nos saca del bucle, pero si retorna true, el bucle sigue
643     for(int i = 0; i < tamaoMax; i++)
644         for(int j = 0; j < tamaoMax; j++)
645             if(tablero[i][j] != null)
646                 if(tablero[i][j].getNombre().equals(texto) && tablero[i][j].getVive()){
647                     if(ejercito != tablero[i][j].getEjercito()){
648                         System.out.println("Los EJERCITOS NO COINCIDEN, por favor ingrese nuevos
                            datos");
649                         return true;
```

```
650         }
651         Imprimir(tablero[i][j]);
652         return false;
653     }
654     System.out.println("Soldado no encontrado, por favor ingrese un nombre valido");
655     return true;
656 }
657
658 public static boolean SumarNiveles(Soldado[][] tablero, int ejercito){
659     //si retorna false, nos saca del bucle, pero si retorna true, el bucle sigue
660     Scanner sc = new Scanner(System.in);
661     Soldado soldado = new Soldado("SumaDeNiveles");
662     for(int i = 0; i < tamaoMax; i++){
663         for(int j = 0; j < tamaoMax; j++){
664             if(tablero[i][j] != null)
665                 if(tablero[i][j].getVive() && tablero[i][j].getEjercito() == ejercito){
666                     soldado.setVidaActual(soldado.getVidaActual() +
667                         tablero[i][j].getVidaActual());
668                     soldado.setAtaque(soldado.getAtaque() + tablero[i][j].getAtaque());
669                     soldado.setDefensa(soldado.getDefensa() + tablero[i][j].getDefensa());
670                     soldado.setVelocidad(soldado.getVelocidad() + tablero[i][j].getVelocidad());
671                 }
672             System.out.println("Desea ver los datos del soldado creado?\n1. SI\n2. NO");
673             int opcion = sc.nextInt();
674             sc.nextLine();
675             if(opcion == 1) Imprimir(soldado);
676             return false;
677         }
678     }
679
680 public static void main(String[] args){
681     Scanner sc = new Scanner(System.in);
682     boolean condicion1 = true;
683     while(condicion1){ // El bucle termina cuando se eliga la opcion 3. SALIR
684         boolean condicion2 = true;
685         System.out.println("\n**** ----- MENU DEL JUEGO ----- ****\n\n" +
686             "Escriba el numero de la opcion que desea elegir\n" +
687             "1. JUEGO RAPIDO\n2. JUEGO PERSONALIZADO\n3. SALIR");
688         int opcion = sc.nextInt();
689         if (opcion == 1){
690             while(condicion2){ // El bucle termina cuando se eliga la opcion 2. SALIR AL MENU
691                 Jugar();
692                 System.out.println("Seleccione:\n1. NUEVO JUEGO\n2. SALIR AL MENU");
693                 opcion = sc.nextInt();
694                 if(opcion == 1){
695                     condicion2 = true; // Se mantiene en el bucle while
696                 } else if (opcion == 2) {
697                     condicion2 = false; // Sale del bucle while
698                 }
699             }
700         } else if (opcion == 2){
701             int tamao1 = (int)(Math.random()* tamaoMax + 1); // Soldados del ejercito 1
702             int tamao2 = (int)(Math.random()* tamaoMax + 1); // Soldados del ejercito 2
703             Soldado[][] tablero = new Soldado[tamaoMax][tamaoMax]; // Creando un arreglo
704                 bidimensional de Soldados
705
706             AsignarAtributos(tablero, tamao1, 1); // Creando Soldados en el ejercito 0
```



```
704     AsignarAtributos(tablero, tamao2, 2); // Creando Soldados en el ejercito 1
705
706     ImprimirTablero(tablero);
707
708     int ejercito = 0;
709
710     while(condicion2){ // El bucle termina cuando se eligan las opciones del 1 al 11
711         boolean condicion3 = true, condicion4 = true;
712         while(condicion3){ // El bucle termina cuando se eliga un ejercito valido, 1
713             0 2
714             System.out.println("\nSeleccione el ejercito al cual quiere modificar\n");
715             ejercito = sc.nextInt();
716             if(ejercito == 1 || ejercito == 2){
717                 System.out.println("\nSe ha seleccionado el ejercito " + ejercito +
718                     "\n");
719                 condicion3 = false;
720             } else System.out.println("\nERROR, Ejercito NO VALIDO. Ingrese los
721                 valores otra vez\n");
722         }
723         System.out.println("Seleccione la accion que desea realizar:\n\n" +
724             "1. CREAR SOLDADO\n2. ELIMINAR SOLDADO\n3. CLONAR SOLDADO\n4. MODIFICAR
725             SOLDADO\n" +
726             "5. COMPARAR SOLDADOS\n6. INTERCAMBIAR SOLDADOS\n7. VER SOLDADO\n8. VER
727             EJERCITO\n" +
728             "9. SUMAR NIVELES\n10. JUGAR\n11. VOLVER\n");
729         opcion = sc.nextInt();
730         String texto;
731         String[] nombres = new String[2];
732         if(opcion == 1){
733             sc.nextLine();
734             if(SoldadosVivos(tablero, ejercito) >= 10){
735                 condicion4 = false;
736                 System.out.println("EL ejercito elegido no puede tener mas de 10
737                     soldados, no se pueden crear mas");
738             }
739             while(condicion4){ // El bucle termina cuando se ESCRIBAN DATOS VALIDOS
740                 System.out.println("Ingrese los datos del soldado de esta manera:" +
741                     "(Ejm. nombre,vida,ejercito,fila,columna,ataque,defensa)");
742                 texto = sc.nextLine();
743                 condicion4 = CrearSoldado(texto, tablero, ejercito);
744             }
745         } else if (opcion == 2) {
746             sc.nextLine();
747             if(SoldadosVivos(tablero, ejercito) < 1){
748                 condicion4 = false;
749                 System.out.println("EL ejercito no tiene soldados vivos, no se pueden
750                     eliminar mas");
751             }
752         }
753         System.out.println("Desea ver los datos de los soldados del tablero?\n1.
754             SI\n2. NO");
755         opcion = sc.nextInt();
756         sc.nextLine();
757         if(opcion == 1) ImprimirDatos(tablero);
758         while(condicion4){ // El bucle termina cuando se ESCRIBAN DATOS VALIDOS
759             System.out.println("Ingrese el nombre del soldado a eliminar");
760             texto = sc.nextLine();
```



```
752         condicion4 = EliminarSoldado(texto, tablero, ejercito);
753     }
754 } else if (opcion == 3) {
755     sc.nextLine();
756     if(SoldadosVivos(tablero, ejercito) < 1){
757         condicion4 = false;
758         System.out.println("EL ejercito no tiene soldados vivos, no se puede
759                             clonar");
760     }
761     System.out.println("Desea ver los datos de los soldados del tablero?\n1.
762                         SI\n2. NO");
763     opcion = sc.nextInt();
764     sc.nextLine();
765     if(opcion == 1) ImprimirDatos(tablero);
766     while(condicion4){ // El bucle termina cuando se ESCRIBAN DATOS VALIDOS
767         System.out.println("Ingrese el nombre del soldado a clonar");
768         texto = sc.nextLine();
769         condicion4 = ClonarSoldado(texto, tablero, ejercito);
770     }
771     System.out.println("Desea ver los datos de los soldados del tablero?\n1.
772                         SI\n2. NO");
773     opcion = sc.nextInt();
774     sc.nextLine();
775     if(opcion == 1) ImprimirDatos(tablero);
776 } else if (opcion == 4) {
777     sc.nextLine();
778     if(SoldadosVivos(tablero, ejercito) < 1){
779         condicion4 = false;
780         System.out.println("EL ejercito no tiene soldados vivos, no se pueden
781                             modificar");
782     }
783     System.out.println("Desea ver los datos de los soldados del tablero?\n1.
784                         SI\n2. NO");
785     opcion = sc.nextInt();
786     sc.nextLine();
787     if(opcion == 1) ImprimirDatos(tablero);
788     while(condicion4){ // El bucle termina cuando se ESCRIBAN DATOS VALIDOS
789         System.out.println("Ingrese el nombre del soldado a modificar");
790         texto = sc.nextLine();
791         condicion4 = ModificarSoldado(texto, tablero, ejercito);
792     }
793     System.out.println("Desea ver los datos de los soldados del tablero?\n1.
794                         SI\n2. NO");
795     opcion = sc.nextInt();
796     sc.nextLine();
797     if(opcion == 1) ImprimirDatos(tablero);
798 } else if (opcion == 5) {
799     sc.nextLine();
800     if(SoldadosVivos(tablero, ejercito) < 2){
801         condicion4 = false;
802         System.out.println("EL ejercito no tiene los soldados necesarios para
803                             comparar sus datos");
804     }
805     System.out.println("Desea ver los datos de los soldados del tablero?\n1.
806                         SI\n2. NO");
807     opcion = sc.nextInt();
```

```
800         sc.nextLine();
801         if(opcion == 1) ImprimirDatos(tablero);
802         while(condicion4){ // El bucle termina cuando se ESCRIBAN DATOS VALIDOS
803             System.out.println("Ingrese el nombre del primer soldado");
804             nombres[0] = sc.nextLine();
805             System.out.println("Ingrese el nombre del segundo soldado");
806             nombres[1] = sc.nextLine();
807             condicion4 = CompararSoldados(nombres, tablero, ejercito);
808         }
809         System.out.println("Desea ver los datos de los soldados del tablero?\n1.
810             SI\n2. NO");
811         opcion = sc.nextInt();
812         sc.nextLine();
813         if(opcion == 1) ImprimirDatos(tablero);
814     } else if (opcion == 6) {
815         sc.nextLine();
816         if(SoldadosVivos(tablero, ejercito) < 2){
817             condicion4 = false;
818             System.out.println("EL ejercito no tiene los soldados necesarios para
819                 intercambiar sus posiciones");
820         }
821         System.out.println("Desea ver los datos de los soldados del tablero?\n1.
822             SI\n2. NO");
823         opcion = sc.nextInt();
824         sc.nextLine();
825         if(opcion == 1) ImprimirDatos(tablero);
826         while(condicion4){ // El bucle termina cuando se ESCRIBAN DATOS VALIDOS
827             System.out.println("Ingrese el nombre del primer soldado");
828             nombres[0] = sc.nextLine();
829             System.out.println("Ingrese el nombre del segundo soldado");
830             nombres[1] = sc.nextLine();
831             condicion4 = IntercambiarSoldados(nombres, tablero, ejercito);
832         }
833         System.out.println("Desea ver los datos de los soldados del tablero?\n1.
834             SI\n2. NO");
835         opcion = sc.nextInt();
836         sc.nextLine();
837         if(opcion == 1) ImprimirDatos(tablero);
838     } else if (opcion == 7) {
839         sc.nextLine();
840         if(SoldadosVivos(tablero, ejercito) < 1){
841             condicion4 = false;
842             System.out.println("EL ejercito no tiene soldados vivos");
843         }
844         while(condicion4){ // El bucle termina cuando se ESCRIBAN DATOS VALIDOS
845             System.out.println("Ingrese el nombre del soldado a imprimir");
846             texto = sc.nextLine();
847             condicion4 = ImprimirSoldado(texto, tablero, ejercito);
848         }
849         System.out.println("Desea ver los datos de los soldados del tablero?\n1.
850             SI\n2. NO");
851         opcion = sc.nextInt();
852         sc.nextLine();
853         if(opcion == 1) ImprimirDatos(tablero);
854     } else if (opcion == 8) {
```

```
851         sc.nextLine();
852         if(SoldadosVivos(tablero, ejercito) < 1){
853             condicion4 = false;
854             System.out.println("EL ejercito no tiene soldados vivos");
855         }
856         ImprimirTablero(tablero, ejercito);
857         System.out.println("Desea ver los datos de los soldados del tablero?\n1.
858             SI\n2. NO");
859         opcion = sc.nextInt();
860         sc.nextLine();
861         if(opcion == 1) ImprimirDatos(tablero);
862     } else if (opcion == 9) {
863         sc.nextLine();
864         if(SoldadosVivos(tablero, ejercito) < 1){
865             condicion4 = false;
866             System.out.println("EL ejercito no tiene soldados vivos, no se puede
867                 trabajar");
868         }
869         System.out.println("Desea ver los datos de los soldados del tablero?\n1.
870             SI\n2. NO");
871         opcion = sc.nextInt();
872         sc.nextLine();
873         if(opcion == 1) ImprimirDatos(tablero);
874         while(condicion4){ // El bucle termina cuando se ESCRIBAN DATOS VALIDOS
875             condicion4 = SumarNiveles(tablero, ejercito);
876         }
877     } else if (opcion == 10) {
878         sc.nextLine();
879         condicion4 = ActualizarCondicion(tablero);
880         if(!condicion4){
881             System.out.println("EL ejercito no tiene soldados vivos, no se puede
882                 jugar");
883         }
884         while(condicion4){ // El bucle termina cuando se ESCRIBAN DATOS VALIDOS
885             ImprimirTablero(tablero);
886             ImprimirDatos(tablero);
887             Jugada(tablero, 1);
888             condicion4 = false;
889         }
890     } else if (opcion == 11) {
891         condicion2 = false; // Sale del bucle while condicion2
892     } else {
893         System.out.println("Opcion NO VALIDA. Ingrese de nuevo");
894         condicion2 = true; // Se mantiene en el bucle while
895     }
896 }
897 }
898 }
899 }
900 }
901 }
```

- **Commit 2:** No se modificaron metodos del constructor Soldado

Listing 2: Clase Persona

```
1 public class Soldado{
2     private String nombre;
3     private int nivelVida;
4     private int vidaActual;
5     private boolean vive;
6     private int ejercito;
7     private int fila;
8     private int columna;
9     private int nivelAtaque;
10    private int nivelDefensa;
11    private int velocidad;
12    private String actitud;
13
14    public Soldado(){
15        nivelVida = 0;
16        vidaActual = 0;
17        nivelAtaque = 0;
18        nivelDefensa = 0;
19        vive = true;
20    }
21
22    public Soldado(String nombre){
23        this.nombre = nombre;
24        nivelVida = 0;
25        vidaActual = 0;
26        nivelAtaque = 0;
27        nivelDefensa = 0;
28        vive = true;
29    }
30
31    public Soldado(String nombre, int nivelVida, int ejercito, int fila, int columna){
32        this.nombre = nombre;
33        this.nivelVida = nivelVida;
34        this.vidaActual = nivelVida;
35        this.ejercito = ejercito;
36        this.fila = fila;
37        this.columna = columna;
38        vive = true;
39    }
40
41    public Soldado(String nombre, int nivelVida, int ejercito, int fila, int columna, int
        nivelAtaque, int nivelDefensa){
42        this.nombre = nombre;
43        this.nivelVida = nivelVida;
44        this.vidaActual = nivelVida;
45        this.ejercito = ejercito;
46        this.fila = fila;
47        this.columna = columna;
48        this.nivelAtaque = nivelAtaque;
49        this.nivelDefensa = nivelDefensa;
50        vive = true;
51    }
```

```
52
53     public void atacar(){
54         avanzar();
55         actitud = "ofensiva";
56     }
57
58     public void defender(){
59         velocidad = 0;
60         actitud = "defensiva";
61     }
62
63     public void avanzar(){
64         velocidad++;
65     }
66
67     public void retroceder(){
68         if(velocidad > 0){
69             velocidad = 0;
70             actitud = "defensiva";
71         } else velocidad --;
72     }
73
74     public void serAtacado(int num){
75         vidaActual -= num;
76     }
77
78     public void huir(){
79         velocidad += 2;
80         actitud = "fuga";
81     }
82
83     public void morir(){
84         vidaActual = 0;
85         vive = false;
86     }
87
88     public void vivir(){
89         vive = true;
90     }
91
92     public void setNombre(String nombre){
93         this.nombre = nombre;
94     }
95
96     public void setVidaActual(int vidaActual){
97         this.vidaActual = vidaActual;
98     }
99
100    public void setEjercito(int ejercito){
101        this.ejercito = ejercito;
102    }
103
104    public void setFila(int fila){
105        this.fila = fila;
106    }
107
```

```
108 public void setColumna(int columna){
109     this.columna = columna;
110 }
111
112 public void setAtaque(int nivelAtaque){
113     this.nivelAtaque = nivelAtaque;
114 }
115
116 public void setDefensa(int nivelDefensa){
117     this.nivelDefensa = nivelDefensa;
118 }
119
120 public void setVelocidad(int velocidad){
121     this.velocidad = velocidad;
122 }
123
124 public String getNombre(){
125     return nombre;
126 }
127
128 public int getVidaActual(){
129     return vidaActual;
130 }
131
132 public int getFila(){
133     return fila;
134 }
135
136 public int getEjercito(){
137     return ejercito;
138 }
139
140 public int getColumna(){
141     return columna;
142 }
143
144 public int getAtaque(){
145     return nivelAtaque;
146 }
147
148 public int getDefensa(){
149     return nivelDefensa;
150 }
151
152 public int getVelocidad(){
153     return velocidad;
154 }
155
156 public boolean getVive(){
157     return vive;
158 }
159 }
```

- **Commit 3:** El diagrama UML no da a conocer todos los aspectos del programa

Soldado	VideoJuego10
<ul style="list-style-type: none"> - String nombre - int nivelVida - int vidaActual - boolean vive - int columna - int fila - int ejercito - int nivelAtaque - int nivelDefensa - int velocidad - String actual 	<ul style="list-style-type: none"> - static int tamañoTablero
<ul style="list-style-type: none"> + Soldado() + Soldado(String nombre) + Soldado(String nombre, int nivelVida, int ejercito, int fila, int columna) + Soldado(String nombre, int nivelVida, int ejercito, int fila, int columna, int nivelAtaque, int nivelDefensa) + void atacar() + void defender() + void avanzar() + void retroceder() + void saltar(int num) + void huir() + void morir() + void setVidaActual(int vidaActual) + void setVidaActual() + void vive() + void setColumna(int columna) + void setFila(int fila) + void setEjercito(int ejercito) + void setAtaque(int nivelAtaque) + void setDefensa(int nivelDefensa) + void setVelocidad(int velocidad) + void getColumna() + void getFila() + String getNombre() + int getEjercito() + boolean getVive() + int getAtaque() + int getDefensa() + int getVelocidad() 	<ul style="list-style-type: none"> + static void Agregar() + static void AgregarRebajas(Soldado[] tablero, int n, int ejercito) + static void ImprimirStatus(Soldado[] tablero) + static void ImprimirTablero(Soldado[] tablero) + static void ImprimirTablero(Soldado[] tablero, int ejercito) + static void Encuentro(Soldado[] tablero, int ejercito) + static void Imprimir(Soldado soldado) + static int PromedioVida(Soldado[] tablero, int ejercito) + static void ImprimirVida(Soldado[] tablero) + static void OrdenamientoPorVida(Soldado[] tablero, int n) + static void OrdenamientoPorEjercito(Soldado[] tablero, int n) + static void Agregar(Soldado[] tablero, int turno) + static int SoldadoVivo(Soldado[] tablero) + static int SoldadoVivo(Soldado[] tablero, int ejercito) + static boolean VerificarSoldado(String soldado, int turno, Soldado[] tablero) + static boolean VerificarDireccion(String direccion, int turno, Soldado[] tablero, String soldado) + static void Combate(Soldado[] tablero, int x, int y, String direccion) + static boolean ActualizarCondiciones(Soldado[] tablero) + static void Ganador(Soldado[] tablero) + static boolean CrearSoldado(String texto, Soldado[] tablero, int ejercitoOriginal) + static boolean EliminarSoldado(String texto, Soldado[] tablero, int ejercito) + static boolean ClonarSoldado(String texto, Soldado[] tablero, int ejercito) + static boolean ModificarSoldado(String texto, Soldado[] tablero, int ejercito) + static boolean CompararSoldado(String nombres, Soldado[] tablero, int ejercito) + static boolean InteraccionarSoldado(String nombres, Soldado[] tablero, int ejercito) + static boolean ImprimirSoldado(String texto, Soldado[] tablero, int ejercito) + static boolean SumarVidas(Soldado[] tablero, int ejercito) + static void menu(String arg)

5. Programa Compilado

Commit 4: Todas las funciones del Juego funcionan correctamente

```
PS C:\Users\jefer> & 'C:\Program Files\Java\jdk-20\bin\java.exe'

**** ----- MENU DEL JUEGO ----- ****

Escriba el numero de la opcion que desea elegir
1. JUEGO RAPIDO
2. JUEGO PERSONALIZADO
3. SALIR
1
Soldado1x3  Salud: 1  Ejercito: 1
Fila:1  Columna: 3
ATQ:4  DEF: 5

Soldado1x2  Salud: 4  Ejercito: 1
Fila:2  Columna: 4
ATQ:2  DEF: 1

Soldado1x8  Salud: 3  Ejercito: 1
Fila:3  Columna: 9
ATQ:1  DEF: 4

Soldado1x6  Salud: 1  Ejercito: 1
Fila:4  Columna: 1
ATQ:3  DEF: 1

Soldado1x5  Salud: 1  Ejercito: 1
Fila:5  Columna: 6
ATQ:1  DEF: 5

Soldado1x1  Salud: 5  Ejercito: 1
Fila:7  Columna: 2
ATQ:1  DEF: 1

Soldado1x7  Salud: 5  Ejercito: 1
Fila:8  Columna: 9
ATQ:4  DEF: 1

Soldado1x9  Salud: 5  Ejercito: 1
Fila:9  Columna: 8
ATQ:3  DEF: 3

Soldado1x4  Salud: 1  Ejercito: 1
Fila:10  Columna: 1
ATQ:1  DEF: 3

Soldado2x1  Salud: 2  Ejercito: 2
Fila:10  Columna: 7
ATQ:2  DEF: 3

Soldado2x2  Salud: 5  Ejercito: 2
Fila:10  Columna: 8
ATQ:5  DEF: 3
```



```
-----  
= = 1 = = = = =  
= = = 1 = = = = =  
= = = = = = 1 =  
1 = = = = = = =  
= = = = = 1 = = =  
= = = = = = = =  
= 1 = = = = = = =  
= = = = = = = 1 =  
= = = = = = 1 = =  
1 = = = = = 2 2 = =
```

```
-----  
  
Soldado con mas vida del ejercito 1 es:  
Soldado1x1 Salud: 5 Ejercito: 1  
Fila:7 Columna: 2  
ATQ:1 DEF: 1
```

```
Soldado con mas vida del ejercito 2 es:  
Soldado2x2 Salud: 5 Ejercito: 2  
Fila:10 Columna: 8  
ATQ:5 DEF: 3
```

```
Promedio de Vida del Ejercito 1: 2
```

```
Promedio de Vida del Ejercito 2: 3
```

```
-----  
  
Vida de todos los soldados:  
Vida de Soldado1x6: 1  
Vida de Soldado1x4: 1  
Vida de Soldado1x1: 5  
Vida de Soldado1x3: 1  
Vida de Soldado1x2: 4  
Vida de Soldado1x5: 1  
Vida de Soldado2x1: 2  
Vida de Soldado1x9: 5  
Vida de Soldado2x2: 5  
Vida de Soldado1x8: 3  
Vida de Soldado1x7: 5  
  
-----
```

Ranking de Vida segun Ordenamiento Burbuja

Soldado1x7 Salud: 5 Ejercito: 1

Fila:8 Columna: 9

ATQ:4 DEF: 1

Soldado2x2 Salud: 5 Ejercito: 2

Fila:10 Columna: 8

ATQ:5 DEF: 3

Soldado1x9 Salud: 5 Ejercito: 1

Fila:9 Columna: 8

ATQ:3 DEF: 3

Soldado1x1 Salud: 5 Ejercito: 1

Fila:7 Columna: 2

ATQ:1 DEF: 1

Soldado1x2 Salud: 4 Ejercito: 1

Fila:2 Columna: 4

ATQ:2 DEF: 1

Soldado1x8 Salud: 3 Ejercito: 1

Fila:3 Columna: 9

ATQ:1 DEF: 4

Soldado2x1 Salud: 2 Ejercito: 2

Fila:10 Columna: 7

ATQ:2 DEF: 3

Soldado1x5 Salud: 1 Ejercito: 1

Fila:5 Columna: 6

ATQ:1 DEF: 5

Soldado1x3 Salud: 1 Ejercito: 1

Fila:1 Columna: 3

ATQ:4 DEF: 5

Soldado1x4 Salud: 1 Ejercito: 1

Fila:10 Columna: 1

ATQ:1 DEF: 3

Soldado1x6 Salud: 1 Ejercito: 1

Fila:4 Columna: 1

ATQ:3 DEF: 1

Ranking de Vida segun Ordenamiento por Insercion

Soldado1x7 Salud: 5 Ejercito: 1

Fila:8 Columna: 9

ATQ:4 DEF: 1

Soldado2x2 Salud: 5 Ejercito: 2

Fila:10 Columna: 8

ATQ:5 DEF: 3

Soldado1x9 Salud: 5 Ejercito: 1

Fila:9 Columna: 8

ATQ:3 DEF: 3

Soldado1x1 Salud: 5 Ejercito: 1

Fila:7 Columna: 2

ATQ:1 DEF: 1

Soldado1x2 Salud: 4 Ejercito: 1

Fila:2 Columna: 4

ATQ:2 DEF: 1

Soldado1x8 Salud: 3 Ejercito: 1

Fila:3 Columna: 9

ATQ:1 DEF: 4

Soldado2x1 Salud: 2 Ejercito: 2

Fila:10 Columna: 7

ATQ:2 DEF: 3

Soldado1x5 Salud: 1 Ejercito: 1

Fila:5 Columna: 6

ATQ:1 DEF: 5

Soldado1x3 Salud: 1 Ejercito: 1

Fila:1 Columna: 3

ATQ:4 DEF: 5

Soldado1x4 Salud: 1 Ejercito: 1

Fila:10 Columna: 1

ATQ:1 DEF: 3

Soldado1x6 Salud: 1 Ejercito: 1

Fila:4 Columna: 1

ATQ:3 DEF: 1

```
-----  
  
Se han creado 11 Soldados
```

```
Si quieres salir al menu principal, coloca *cancelar* en cualquier momento  
Jugador 1. Ingrese el soldado a mover del Ejercito 1. (Ejm. Soldado1x1)  
Soldado1x4
```

```
Ahora ingrese la direccion (w:up, s:down, d:righ, a:left)
```

```
d
```

```
= = 1 = = = = =
```

```
= = = 1 = = = = =
```

```
= = = = = = 1 =
```

```
1 = = = = = = =
```

```
= = = = = 1 = = =
```

```
= = = = = = = =
```

```
= 1 = = = = = =
```

```
= = = = = = 1 =
```

```
= = = = = = 1 =
```

```
= 1 = = = = 2 2 =
```

```
-----
```

```
Soldado1x3  Salud: 1  Ejercito: 1
Fila:1  Columna: 3
ATQ:4  DEF: 5

Soldado1x2  Salud: 4  Ejercito: 1
Fila:2  Columna: 4
ATQ:2  DEF: 1

Soldado1x8  Salud: 3  Ejercito: 1
Fila:3  Columna: 9
ATQ:1  DEF: 4

Soldado1x6  Salud: 1  Ejercito: 1
Fila:4  Columna: 1
ATQ:3  DEF: 1

Soldado1x5  Salud: 1  Ejercito: 1
Fila:5  Columna: 6
ATQ:1  DEF: 5

Soldado1x1  Salud: 5  Ejercito: 1
Fila:7  Columna: 2
ATQ:1  DEF: 1

Soldado1x7  Salud: 5  Ejercito: 1
Fila:8  Columna: 9
ATQ:4  DEF: 1

Soldado1x9  Salud: 5  Ejercito: 1
Fila:9  Columna: 8
ATQ:3  DEF: 3

Soldado1x4  Salud: 1  Ejercito: 1
Fila:10  Columna: 2
ATQ:0  DEF: 0

Soldado2x1  Salud: 2  Ejercito: 2
Fila:10  Columna: 7
ATQ:2  DEF: 3

Soldado2x2  Salud: 5  Ejercito: 2
Fila:10  Columna: 8
ATQ:5  DEF: 3
```

```
Hay 9 Soldados vivos del ejercito 1

Hay 2 Soldados vivos del ejercito 2

Si quieres salir al menu principal, coloca *cancelar* en cualquier momento
Jugador 2. Ingrese el soldado a mover del Ejercito 2. (Ejm. Soldado1x1)
cancelar
Seleccione:
1. NUEVO JUEGO
2. SALIR AL MENU
2

**** ----- MENU DEL JUEGO ----- ****

Escriba el numero de la opcion que desea elegir
1. JUEGO RAPIDO
2. JUEGO PERSONALIZADO
3. SALIR
2
= = = = 2 = = 1 2
= = 2 = = = = =
= = = = = 2 = =
= = 1 = = = = =
= = = = = = =
= = = = = 1 =
= = = = = = =
= = 1 = = = = =
2 = = = = 2 = = =
= = = = = = =

-----

Seleccione el ejercito al cual quiere modificar

1

Se ha seleccionado el ejercito 1

Seleccione la accion que desea realizar:

1. CREAR SOLDADO
2. ELIMINAR SOLDADO
3. CLONAR SOLDADO
4. MODIFICAR SOLDADO
5. COMPARAR SOLDADOS
6. INTERCAMBIAR SOLDADOS
7. VER SOLDADO
8. VER EJERCITO
9. SUMAR NIVELES
10. JUGAR
11. VOLVER
```

```
Hay 9 Soldados vivos del ejercito 1

Hay 2 Soldados vivos del ejercito 2

Si quieres salir al menu principal, coloca *cancelar* en cualquier momento
Jugador 2. Ingrese el soldado a mover del Ejercito 2. (Ejm. Soldado1x1)
cancelar
Seleccione:
1. NUEVO JUEGO
2. SALIR AL MENU
2

**** ----- MENU DEL JUEGO ----- ****

Escriba el numero de la opcion que desea elegir
1. JUEGO RAPIDO
2. JUEGO PERSONALIZADO
3. SALIR
2
= = = = 2 = = 1 2
= = 2 = = = = =
= = = = = 2 = =
= = 1 = = = = =
= = = = = = =
= = = = = = 1 =
= = = = = = =
= = 1 = = = = =
2 = = = = 2 = = =
= = = = = = =
```

Seleccione el ejercito al cual quiere modificar

1

Se ha seleccionado el ejercito 1

Seleccione la accion que desea realizar:

1. CREAR SOLDADO
2. ELIMINAR SOLDADO
3. CLONAR SOLDADO
4. MODIFICAR SOLDADO
5. COMPARAR SOLDADOS
6. INTERCAMBIAR SOLDADOS
7. VER SOLDADO
8. VER EJERCITO
9. SUMAR NIVELES
10. JUGAR
11. VOLVER

8

```
= = = = = 1 =  
= = = = =  
= = = = =  
= = 1 = = = = =  
= = = = =  
= = = = = 1 =  
= = = = =  
= = 1 = = = = =  
= = = = =  
= = = = =
```

Desea ver los datos de los soldados del tablero?

1. SI
2. NO

2

Seleccione el ejercito al cual quiere modificar

2

Se ha seleccionado el ejercito 2

Seleccione la accion que desea realizar:

1. CREAR SOLDADO
2. ELIMINAR SOLDADO
3. CLONAR SOLDADO
4. MODIFICAR SOLDADO
5. COMPARAR SOLDADOS
6. INTERCAMBIAR SOLDADOS
7. VER SOLDADO
8. VER EJERCITO
9. SUMAR NIVELES
10. JUGAR
11. VOLVER

4

Desea ver los datos de los soldados del tablero?

1. SI
2. NO

1

```
Soldado2x4  Salud: 4  Ejercito: 2
Fila:1  Columna: 6
ATQ:1  DEF: 4
```

```
Soldado1x4  Salud: 1  Ejercito: 1
Fila:1  Columna: 9
ATQ:3  DEF: 5
```

```
Soldado2x5  Salud: 2  Ejercito: 2
Fila:1  Columna: 10
ATQ:3  DEF: 1
```

```
Soldado2x3  Salud: 5  Ejercito: 2
Fila:2  Columna: 3
ATQ:5  DEF: 2
```

```
Soldado2x1  Salud: 2  Ejercito: 2
Fila:3  Columna: 8
ATQ:4  DEF: 5
```

```
Soldado1x2  Salud: 5  Ejercito: 1
Fila:4  Columna: 3
ATQ:2  DEF: 4
```

```
Soldado1x1  Salud: 4  Ejercito: 1
Fila:6  Columna: 9
ATQ:1  DEF: 3
```

```
Soldado1x3  Salud: 4  Ejercito: 1
Fila:8  Columna: 3
ATQ:2  DEF: 4
```

```
Soldado2x6  Salud: 1  Ejercito: 2
Fila:9  Columna: 1
ATQ:2  DEF: 3
```

```
Soldado2x2  Salud: 1  Ejercito: 2
Fila:9  Columna: 6
ATQ:4  DEF: 5
```

```
-----  
Ingrese el nombre del soldado a modificar  
Soldado2x4  
Seleccione que quiere modificar:  
1. VIDA  
2. ATAQUE  
3. DEFENSA  
4. SALIR  
  
4  
Desea ver los datos de los soldados del tablero?  
1. SI  
2. NO  
2  
  
Seleccione el ejercito al cual quiere modificar  
  
1  
  
Se ha seleccionado el ejercito 1  
  
Seleccione la accion que desea realizar:  
  
1. CREAR SOLDADO  
2. ELIMINAR SOLDADO  
3. CLONAR SOLDADO  
4. MODIFICAR SOLDADO  
5. COMPARAR SOLDADOS  
6. INTERCAMBIAR SOLDADOS  
7. VER SOLDADO  
8. VER EJERCITO  
9. SUMAR NIVELES  
10. JUGAR  
11. VOLVER  
  
10  
= = = = 2 = = 1 2  
= = 2 = = = = =  
= = = = = = 2 = =  
= = 1 = = = = =  
= = = = = = = =  
= = = = = = = 1 =  
= = = = = = = =  
= = 1 = = = = =  
2 = = = = 2 = = =  
= = = = = = = =
```

```
-----  
Soldado2x4  Salud: 4  Ejercito: 2  
Fila:1  Columna: 6  
ATQ:1  DEF: 4
```

```
Soldado1x4  Salud: 1  Ejercito: 1  
Fila:1  Columna: 9  
ATQ:3  DEF: 5
```

```
Soldado2x5  Salud: 2  Ejercito: 2  
Fila:1  Columna: 10  
ATQ:3  DEF: 1
```

```
Soldado2x3  Salud: 5  Ejercito: 2  
Fila:2  Columna: 3  
ATQ:5  DEF: 2
```

```
Soldado2x1  Salud: 2  Ejercito: 2  
Fila:3  Columna: 8  
ATQ:4  DEF: 5
```

```
Soldado1x2  Salud: 5  Ejercito: 1  
Fila:4  Columna: 3  
ATQ:2  DEF: 4
```

```
Soldado1x1  Salud: 4  Ejercito: 1  
Fila:6  Columna: 9  
ATQ:1  DEF: 3
```

```
Soldado1x3  Salud: 4  Ejercito: 1  
Fila:8  Columna: 3  
ATQ:2  DEF: 4
```

```
Soldado2x6  Salud: 1  Ejercito: 2  
Fila:9  Columna: 1  
ATQ:2  DEF: 3
```

```
Soldado2x2  Salud: 1  Ejercito: 2  
Fila:9  Columna: 6  
ATQ:4  DEF: 5
```

```
-----  
Se han creado 10 Soldados
```

```
Si quieres salir al menu principal, coloca *cancelar* en cualquier momento  
Jugador 1. Ingrese el soldado a mover del Ejercito 1. (Ejm. Soldado1x1)
```

6. Referencias

- https://www.w3schools.com/js/js_ajax_intro.asp
- <https://www.youtube.com/watch?v=cAqmF7mtZv0>

- <https://www.youtube.com/watch?v=cAqmF7mtZv0>