

Informe de Laboratorio 10

Tema: Definición de Clases de Usuario Clase Soldado

Nota

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

Laboratorio	Tema	Duración
10	Definición de Clases de Usuario Clase Soldado	48 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - B		

Docente
Aedo Lopez, Marco Wilfredo

1. Tarea

- Que el alumno demuestre poder crear “clases definidas por el programador”
- Implementar métodos para las clases definidas por el programador
- Crear Métodos Sobrecargados (Overloaded)

2. Equipos, Materiales y temas utilizados

- Sistema Operativo Windows 11.
- VIM 9.0.
- OpenJDK 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.
- https://github.com/JefersonSH/FP2-2024B/tree/eb261ca92998cb186b738e515a242259f00efc83/Laboratorio_10

4. Ejercicios Resuletos

4.1. Commits

Ejercicio 01

- **Commit 1:** VideoJuego numero 7 con las modificaciones pedidas por el docente

Listing 1: Clase Direccion

```
1 import java.util.*;
2
3 public class VideoJuego7 {
4
5     public static void main(String[] args){
6         Scanner sc = new Scanner(System.in);
7         boolean condicion = true;
8
9         int tamao1 = (int)(Math.random()*10+1); //Cantidad de soldados del ejercito 1
10        int tamao2 = (int)(Math.random()*10+1); //Cantidad de soldados del ejercito 2
11        Soldado[][] tablero = new Soldado[10][10]; //Creando un arreglo de Soldados
12                                   bidimensional
13
14        AsignarAtributos(tablero, tamao1, 1); //Creando Soldados en el ejercito 0
15        AsignarAtributos(tablero, tamao2, 2); //Creando Soldados en el ejercito 1
16
17        ImprimirDatos(tablero);
18        ImprimirTablero(tablero);
19
20        System.out.println("El soldado con mas vida del ejercito 1 es:");
21        EncontrarMayorVida(tablero, 10, 1);
22        System.out.println("El soldado con mas vida del ejercito 2 es:");
23        EncontrarMayorVida(tablero, 10, 2);
24
25        System.out.println("Promedio de Vida del Ejercito 1: " + PromedioVida(tablero, 10, 1) +
26                               "\n");
27        System.out.println("Promedio de Vida del Ejercito 2: " + PromedioVida(tablero, 10, 2) +
28                               "\n");
29        System.out.println("\n-----\n");
30
31        System.out.println("Vida de todos los soldados:");
32        ImprimirVida(tablero);
33
34        System.out.println("Ranking de Vida segun Ordenamiento Burbuja");
35        OrdenamientoBurbuja(tablero, SoldadosVivos(tablero));
```

```
33 System.out.println("Ranking de Vida segun Ordenamiento por Insercion");
34 OrdenamientoInsercion(tablero, SoldadosVivos(tablero));
35
36 int turno = 1;
37
38 while(condicion){
39     Jugada(tablero, turno); //Este metodo ejecutara toda la jugada de cada turno
40     ImprimirTablero(tablero); //Imprimir el tablero y datos despues de cada turno
41     ImprimirDatos(tablero);
42
43     condicion = ActualizarCondicion(tablero); //Si hay un ejercito con 0 soldados, sera
44         false
45     turno++;
46 }
47 Ganador(tablero);
48 }
49
50 public static void AsignarAtributos(Soldado[] [] tablero, int n, int ejercito){
51     for(int i = 0; i < n; i++){
52         int x = (int)(Math.random()*10); //x representa a las filas, siendo del 1 al 9
53         int y = (int)(Math.random()*10); //y representa a las columnas, siendo del 1 al 9
54         if(tablero[x][y] == null){ //Si la posicion [y][x] esta vacia, crea un Soldado en
55             esa posicion
56             String nombre = "Soldado" + (ejercito) + "x" + (i + 1);
57             int vida = (int)(Math.random()*5+1);
58             int ataque = (int)(Math.random()*5+1);
59             int defensa = (int)(Math.random()*5+1);
60             tablero[x][y] = new Soldado(nombre, vida, ejercito, x + 1, y + 1, ataque, defensa);
61         } else {
62             i--; //Si no hay un Soldado, se vuelve al ciclo
63         }
64     }
65 }
66
67 public static void ImprimirDatos(Soldado[] [] tablero){
68     for(int i = 0; i < 10; i++) //Controla las filas
69         for(int j = 0; j < 10; j++) //Controla las columnas
70             if(tablero[i][j] != null)
71                 if(tablero[i][j].getVive()){
72                     System.out.println("Nombre: " + tablero[i][j].getNombre());
73                     System.out.println("Salud: " + tablero[i][j].getVidaActual());
74                     System.out.println("Ejercito: " + tablero[i][j].getEjercito());
75                     System.out.println("Fila: " + tablero[i][j].getFila());
76                     System.out.println("Columna: " + tablero[i][j].getColumna());
77                 }
78     System.out.println("\n-----\n");
79 }
80
81 public static void ImprimirTablero(Soldado[] [] tablero){
82     for(int i = 0; i < 10; i++){ //filas
83         for(int j = 0; j < 10; j++){ //columnas
84             if(tablero[i][j] != null) //Si el indice contiene un Soldado continua
85                 if(tablero[i][j].getVive()){ //Si el soldado esta vivo continua
86                     if(tablero[i][j].getEjercito() == 1) System.out.print("1 ");
87                     if(tablero[i][j].getEjercito() == 2) System.out.print("2 ");
88                 } else System.out.print("= ");
```

```
87         else System.out.print("= ");
88
89     }
90     System.out.print("\n"); //Una vez acabada una fila, se pasa a la siguiente linea
91 }
92 System.out.println("\n-----\n");
93 }
94
95 public static void EncontrarMayorVida(Soldado[] [] tablero, int n, int ejercito){
96     int mayor = Integer.MIN_VALUE;
97
98     Soldado masVida = new Soldado();
99     for(int i = 0; i < n; i++) //Filas
100         for(int j = 0; j < n; j++) //Columnas
101             if(tablero[j][i] != null)
102                 if(tablero[j][i].getVive()){
103                     if(tablero[j][i].getEjercito() == ejercito) //Para solamente comparar
104                         soldados del mismo ejercito
105                         if(tablero[j][i].getVidaActual() > mayor){
106                             mayor = tablero[j][i].getVidaActual(); //Si es mayor, actualiza la
107                             variable
108                             masVida = tablero[j][i];
109                         }
110                 }
111     System.out.println("Nombre: " + masVida.getNombre());
112     System.out.println("Vida: " + masVida.getVidaActual());
113     System.out.println("Fila: " + masVida.getFila());
114     System.out.println("Fila: " + masVida.getFila());
115     System.out.println("Columna: " + masVida.getColumna());
116     System.out.println("Ejercito" + masVida.getEjercito());
117     System.out.println("\n-----\n");
118 }
119
120 public static int PromedioVida(Soldado[] [] tablero, int n, int ejercito){
121     int promedio = 0;
122     int size = 0;
123     for(int i = 0; i < n; i++) //Filas
124         for(int j = 0; j < n; j++) //Columnas
125             if(tablero[j][i] != null)
126                 if(tablero[j][i].getVive() && tablero[j][i].getEjercito() == ejercito){
127                     promedio += tablero[j][i].getVidaActual();
128                     size++;
129                 }
130     return promedio/size;
131 }
132
133 public static void ImprimirVida(Soldado[] [] tablero){
134     for(int i = 0; i < 10; i++) //Controla las filas
135         for(int j = 0; j < 10; j++) //Controla las columnas
136             if(tablero[j][i] != null)
137                 if(tablero[j][i].getVive())
138                     System.out.println("Vida de " + tablero[j][i].getNombre() + ": " +
139                         tablero[j][i].getVidaActual());
140     System.out.println("\n-----\n");
141 }
```

```
140 public static int SoldadosVivos(Soldado[] [] tablero){
141     int contador = 0;
142     for(int i = 0; i < 10; i++) //Controla las filas
143         for(int j = 0; j < 10; j++) //Controla las columnas
144             if(tablero[i][j] != null)
145                 if(tablero[i][j].getVive())
146                     contador++;
147     return contador;
148 }
149
150 public static int SoldadosVivos(Soldado[] [] tablero, int ejercito){
151     int contador = 0;
152     for(int i = 0; i < 10; i++) //Controla las filas
153         for(int j = 0; j < 10; j++) //Controla las columnas
154             if(tablero[i][j] != null)
155                 if(tablero[i][j].getVive() && tablero[i][j].getEjercito() == ejercito)
156                     contador++;
157     return contador;
158 }
159
160 public static void OrdenamientoBurbuja(Soldado[] [] tablero, int n){
161     Soldado temp = new Soldado();
162     int contador = 0;
163     Soldado[] lista = new Soldado[n];
164     for(int i = 0; i < 10; i++) //Controla las filas
165         for(int j = 0; j < 10; j++) //Controla las columnas
166             if(tablero[j][i] != null){
167                 if(tablero[j][i].getVive()){
168                     lista[contador] = tablero[j][i];
169                     contador++;
170                 }
171             }
172
173     for(int i = 0; i < n - 1; i++)
174         for(int j = 0; j < n - 1; j++)
175             if(lista[j].getVidaActual() > lista[j+1].getVidaActual()){
176                 temp = lista[j+1];
177                 lista[j+1] = lista[j];
178                 lista[j] = temp;
179             }
180     for(int i = n - 1; i >= 0; i--){
181         System.out.println("Nombre: " + lista[i].getNombre());
182         System.out.println("Vida: " + lista[i].getVidaActual());
183         System.out.println("Fila: " + lista[i].getFila());
184         System.out.println("Columna" + lista[i].getColumna());
185         System.out.println("Ejercito" + lista[i].getEjercito());
186     }
187     System.out.println("\n-----\n");
188 }
189
190 public static void OrdenamientoInsercion(Soldado[] [] tablero, int n){
191     Soldado temp = new Soldado();
192     int contador = 0;
193     Soldado[] lista = new Soldado[n];
194     for(int i = 0; i < 10; i++) //Controla las filas
195         for(int j = 0; j < 10; j++) //Controla las columnas
```

```
196         if(tablero[j][i] != null){
197             if(tablero[j][i].getVive()){
198                 lista[contador] = tablero[j][i];
199                 contador++;
200             }
201         }
202
203     for(int i = 1; i < n; i++){
204         Soldado key = lista[i];
205         int j = i - 1;
206         while(j >= 0 && lista[j].getVidaActual() > key.getVidaActual()){
207             lista[j + 1] = lista[j];
208             j = j - 1;
209         }
210         lista[j + 1] = key;
211     }
212
213     for(int i = n - 1; i >= 0; i--){
214         System.out.println("Nombre: " + lista[i].getNombre());
215         System.out.println("Vida: " + lista[i].getVidaActual());
216         System.out.println("Fila: " + lista[i].getFila());
217         System.out.println("Columna" + lista[i].getColumna());
218         System.out.println("Ejercito" + lista[i].getEjercito());
219     }
220     System.out.println("\n-----\n");
221 }
222
223 public static void Jugada(Soldado[] [] tablero, int turno){
224     Scanner sc = new Scanner(System.in);
225     turno %= 2; //Solo habra 2 turnos, turnos pares e impares
226     if(turno == 0) turno = 2;
227     //Pidiendo al usuario valores y verificando que sean correctos
228     System.out.println("Jugador " + turno + ". Ingrese el soldado a mover(Soldado1x1) del
229         Ejercito " + turno);
230     String soldado = sc.nextLine();
231     boolean condicion1 = VerificarSoldado(soldado, turno, tablero);
232     while(!condicion1){
233         System.out.println("Soldado no encontrado, por favor ingrese otro");
234         soldado = sc.nextLine();
235         condicion1 = VerificarSoldado(soldado, turno, tablero);
236     }
237
238     System.out.println("Ahora ingrese la direccion (w:up, s:down, d:righth, a:left)");
239     String direccion = sc.nextLine();
240     boolean condicion2 = VerificarDireccion(direccion, turno, tablero, soldado);
241     while(!condicion2){
242         System.out.println("Direccion no admitida, por favor ingrese otro");
243         direccion = sc.nextLine();
244         condicion2 = VerificarDireccion(direccion, turno, tablero, soldado);
245     }
246     //Llevando a cabo la batalla en caso hubiese
247     SoldadoMovriendose(tablero, turno, soldado, direccion);
248
249     public static boolean VerificarSoldado(String soldado, int turno, Soldado[] [] tablero){
250         for(int i = 0; i < 10; i++) //Controla las filas
```

```
251     for(int j = 0; j < 10; j++) //Controla las columnas
252         if(tablero[i][j] != null)
253             if(tablero[i][j].getVive() && tablero[i][j].getEjercito() == turno)
254                 if(tablero[i][j].getNombre().equals(soldado))
255                     return true;
256     return false;
257 }
258
259 public static boolean VerificarDireccion(String direccion, int turno, Soldado[][] tablero,
260     String soldado){
261     if(!(direccion.equals("w") || direccion.equals("s") || direccion.equals("a") ||
262         direccion.equals("d"))
263         return false;
264
265     for(int i = 0; i < 10; i++) { //Controla las filas
266         for(int j = 0; j < 10; j++) { //Controla las columnas
267             if(tablero[i][j] != null && tablero[i][j].getVive() && tablero[i][j].getEjercito()
268                 == turno && tablero[i][j].getNombre().equals(soldado)){
269                 if(direccion.equals("w") && i - 1 >= 0) // Verifica movimiento hacia arriba
270                     return true;
271                 else if(direccion.equals("s") && i + 1 < 10) // Verifica movimiento hacia abajo
272                     return true;
273                 else if(direccion.equals("a") && j - 1 >= 0) // Verifica movimiento hacia la
274                     izquierda
275                     return true;
276                 else if(direccion.equals("d") && j + 1 < 10) // Verifica movimiento hacia la
277                     derecha
278                     return true;
279             }
280         }
281     }
282     return false;
283 }
284
285 public static void SoldadoMovriendose(Soldado[][] tablero, int turno, String soldado,
286     String direccion){
287     int fila, columna;
288     int contador = 0; //Este contador es MUY IMPORTANTE, evitara que se llame el metodo
289     Combate() mas de 1 vez, evitando errores con s y d
290     for(int i = 0; i < 10; i++) //Controla las filas
291         for(int j = 0; j < 10; j++) //Controla las columnas
292             if(tablero[i][j] != null && tablero[i][j].getVive() && tablero[i][j].getEjercito()
293                 == turno && tablero[i][j].getNombre().equals(soldado) && contador == 0){
294                 if(direccion.equals("w") && i - 1 >= 0){ // Verifica movimiento hacia arriba
295                     Combate(tablero, i, j, direccion);
296                     contador++;
297                 }else if(direccion.equals("s") && i + 1 < 10){ // Verifica movimiento hacia
298                     abajo
299                     Combate(tablero, i, j, direccion);
300                     contador++;
301                 }else if(direccion.equals("a") && j - 1 >= 0){ // Verifica movimiento hacia la
302                     izquierda
303                     Combate(tablero, i, j, direccion);
304                     contador++;
305                 }else if(direccion.equals("d") && j + 1 < 10){ // Verifica movimiento hacia la
306                     derecha
```

```
296         Combate(tablero, i, j, direccion);
297         contador++;
298     }
299 }
300
301
302 public static void Combate(Soldado[][] tablero, int x, int y, String direccion){
303     if(direccion.equals("w")){ //La estructura se repite para cada una de las 4 letras,
304         //pero con sus variaciones
305         if(tablero[x - 1][y] == null || !tablero[x - 1][y].getVive()){ //Verifica que no
306             //haya un Soldado vivo en el casillero a avanzar
307             tablero[x - 1][y] = new Soldado(tablero[x][y].getNombre(),
308                 tablero[x][y].getVidaActual(), tablero[x][y].getEjercito(), x + 1 - 1, y + 1);
309             tablero[x][y].morir();
310         } else if(tablero[x][y].getVidaActual() < tablero[x - 1][y].getVidaActual()){
311             //Verifica que el soldado actual tenga menos vida que el soldado enemigo
312             tablero[x - 1][y].setVidaActual(tablero[x - 1][y].getVidaActual() -
313                 tablero[x][y].getVidaActual());
314             tablero[x][y].morir();
315         } else if(tablero[x][y].getVidaActual() == tablero[x - 1][y].getVidaActual()){
316             //Verifica que el soldado actual tenga la misma vida que el soldado enemigo
317             tablero[x][y].morir();
318             tablero[x - 1][y].morir();
319         } else if(tablero[x][y].getVidaActual() > tablero[x - 1][y].getVidaActual()){
320             //Verifica que el soldado actual tenga mas vida que el soldado enemigo
321             tablero[x - 1][y] = new Soldado(tablero[x][y].getNombre(),
322                 tablero[x][y].getVidaActual() - tablero[x - 1][y].getVidaActual(),
323                 tablero[x][y].getEjercito(), x + 1 - 1, y + 1);
324             tablero[x][y].morir();
325         }
326     }
327     if(direccion.equals("s")){
328         if(tablero[x + 1][y] == null || !tablero[x + 1][y].getVive()){
329             tablero[x + 1][y] = new Soldado(tablero[x][y].getNombre(),
330                 tablero[x][y].getVidaActual(), tablero[x][y].getEjercito(), x + 1 + 1, y + 1);
331             tablero[x][y].morir();
332         } else if(tablero[x][y].getVidaActual() < tablero[x + 1][y].getVidaActual()){
333             tablero[x + 1][y].setVidaActual(tablero[x + 1][y].getVidaActual() -
334                 tablero[x][y].getVidaActual());
335             tablero[x][y].morir();
336         } else if(tablero[x][y].getVidaActual() == tablero[x + 1][y].getVidaActual()){
337             tablero[x][y].morir();
338             tablero[x + 1][y].morir();
339         } else if(tablero[x][y].getVidaActual() > tablero[x + 1][y].getVidaActual()){
340             tablero[x + 1][y] = new Soldado(tablero[x][y].getNombre(),
341                 tablero[x][y].getVidaActual() - tablero[x + 1][y].getVidaActual(),
342                 tablero[x][y].getEjercito(), x + 1 + 1, y + 1);
343             tablero[x][y].morir();
344         }
345     }
346     if(direccion.equals("a")){
347         if(tablero[x][y - 1] == null || !tablero[x][y - 1].getVive()){
348             tablero[x][y - 1] = new Soldado(tablero[x][y].getNombre(),
349                 tablero[x][y].getVidaActual(), tablero[x][y].getEjercito(), x + 1, y + 1 - 1);
350             tablero[x][y].morir();
351         } else if(tablero[x][y].getVidaActual() < tablero[x][y - 1].getVidaActual()){
352             tablero[x][y - 1].setVidaActual(tablero[x][y - 1].getVidaActual() -
```



```
337         tablero[x][y].getVidaActual());
338         tablero[x][y].morir();
339     } else if(tablero[x][y].getVidaActual() == tablero[x][y - 1].getVidaActual()){
340         tablero[x][y].morir();
341         tablero[x][y - 1].morir();
342     } else if(tablero[x][y].getVidaActual() > tablero[x][y - 1].getVidaActual()){
343         tablero[x][y - 1] = new Soldado(tablero[x][y].getNombre(),
344             tablero[x][y].getVidaActual() - tablero[x][y - 1].getVidaActual(),
345             tablero[x][y].getEjercito(), x + 1, y + 1 - 1);
346         tablero[x][y].morir();
347     }
348 } else if(direccion.equals("d")){
349     if(tablero[x][y + 1] == null || !tablero[x][y + 1].getVive()){
350         tablero[x][y + 1] = new Soldado(tablero[x][y].getNombre(),
351             tablero[x][y].getVidaActual(), tablero[x][y].getEjercito(), x + 1, y + 1 + 1);
352         tablero[x][y].morir();
353     } else if(tablero[x][y].getVidaActual() < tablero[x][y + 1].getVidaActual()){
354         tablero[x][y + 1].setVidaActual(tablero[x][y + 1].getVidaActual() -
355             tablero[x][y].getVidaActual());
356         tablero[x][y].morir();
357     } else if(tablero[x][y].getVidaActual() == tablero[x][y + 1].getVidaActual()){
358         tablero[x][y].morir();
359         tablero[x][y + 1].morir();
360     } else if(tablero[x][y].getVidaActual() > tablero[x][y + 1].getVidaActual()){
361         tablero[x][y + 1] = new Soldado(tablero[x][y].getNombre(),
362             tablero[x][y].getVidaActual() - tablero[x][y + 1].getVidaActual(),
363             tablero[x][y].getEjercito(), x + 1, y + 1 + 1);
364         tablero[x][y].morir();
365     }
366 }
367 }
368
369 public static boolean ActualizarCondicion(Soldado[][] tablero){
370     int ejercito1 = SoldadosVivos(tablero, 1);
371     int ejercito2 = SoldadosVivos(tablero, 2);
372     if(ejercito1 == 0 || ejercito2 == 0)
373         return false;
374     return true;
375 }
376
377 public static void Ganador(Soldado[][] tablero){
378     if(SoldadosVivos(tablero, 1) == 0)
379         System.out.println("El Ganador es el Ejercito 2");
380     else if(SoldadosVivos(tablero, 2) == 0)
381         System.out.println("El Ganador es el Ejercito 1");
382     else System.out.println("Error");
383 }
```

- **Commit 2:** Se modificaron y crearon algunos metodos del constructor

Listing 2: Clase Persona

```
1 public class Soldado{
2     private String nombre;
```

```
3 private int nivelAtaque;
4 private int nivelDefensa;
5 private int nivelVida;
6 private int vidaActual;
7 private int velocidad;
8 private String actitud;
9 private boolean vive;
10 private int columna;
11 private int fila;
12 private int ejercito;
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, int nivelVida, int ejercito, int fila, int columna){
23     this.nombre = nombre;
24     this.nivelVida = nivelVida;
25     this.vidaActual = nivelVida;
26     this.ejercito = ejercito;
27     this.fila = fila;
28     this.columna = columna;
29     vive = true;
30 }
31
32 public Soldado(String nombre, int nivelVida, int ejercito, int fila, int columna, int
    nivelAtaque, int nivelDefensa){
33     this.nombre = nombre;
34     this.nivelVida = nivelVida;
35     this.vidaActual = nivelVida;
36     this.ejercito = ejercito;
37     this.fila = fila;
38     this.columna = columna;
39     this.nivelAtaque = nivelAtaque;
40     this.nivelDefensa = nivelDefensa;
41     vive = true;
42 }
43
44 public void atacar(){
45     avanzar();
46     actitud = "ofensiva";
47 }
48
49 public void defender(){
50     velocidad = 0;
51     actitud = "defensiva";
52 }
53
54 public void avanzar(){
55     velocidad++;
56 }
57
```

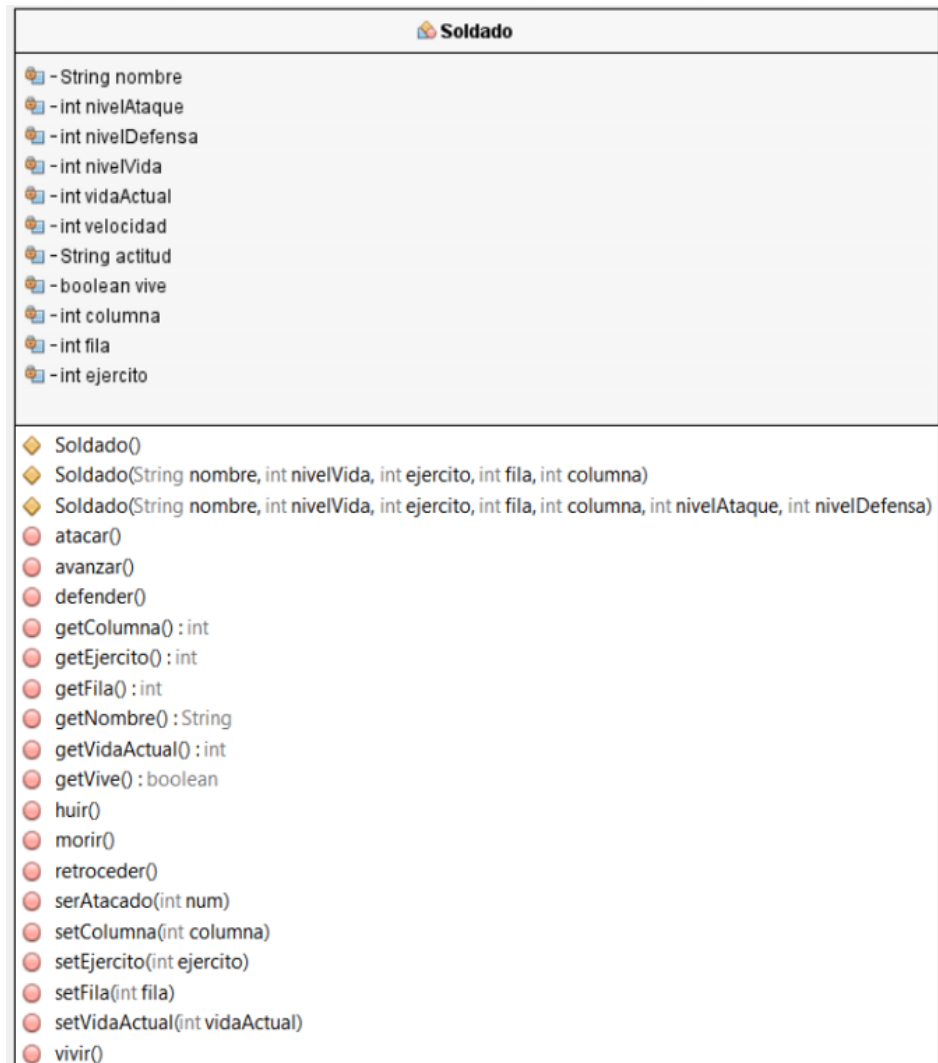
```
58 public void retroceder(){
59     if(velocidad > 0){
60         velocidad = 0;
61         actitud = "defensiva";
62     } else velocidad --;
63 }
64
65 public void serAtacado(int num){
66     vidaActual -= num;
67 }
68
69 public void huir(){
70     velocidad += 2;
71     actitud = "fuga";
72 }
73
74 public void morir(){
75     vidaActual = 0;
76     vive = false;
77 }
78
79 public void vivir(){
80     vive = true;
81 }
82
83 public void setVidaActual(int vidaActual){
84     this.vidaActual = vidaActual;
85 }
86
87 public int getVidaActual(){
88     return vidaActual;
89 }
90
91 public void setColumna(int columna){
92     this.columna = columna;
93 }
94
95 public int getColumna(){
96     return columna;
97 }
98
99 public void setFila(int fila){
100     this.fila = fila;
101 }
102
103 public int getFila(){
104     return fila;
105 }
106
107 public void setEjercito(int ejercito){
108     this.ejercito = ejercito;
109 }
110
111 public int getEjercito(){
112     return ejercito;
113 }
```

```

114
115     public boolean getVive(){
116         return vive;
117     }
118
119     public String getNombre(){
120         return nombre;
121     }
122 }

```

■ Diagrama UML



5. Programa Compilado

```
Nombre: Soldado1x4
Salud: 3
Ejercito: 1
Fila:1
Columna: 3
Nombre: Soldado1x2
Salud: 2
Ejercito: 1
Fila:4
Columna: 8
Nombre: Soldado2x8
Salud: 1
Ejercito: 2
Fila:5
Columna: 1
Nombre: Soldado1x5
Salud: 1
Ejercito: 1
Fila:5
Columna: 7
Nombre: Soldado2x5
Salud: 5
Ejercito: 2
Fila:6
Columna: 9
Nombre: Soldado2x7
Salud: 5
Ejercito: 2
Fila:7
Columna: 3
Nombre: Soldado2x1
Salud: 2
Ejercito: 2
Fila:8
Columna: 2
Nombre: Soldado2x3
Salud: 4
Ejercito: 2
Fila:8
Columna: 4
Nombre: Soldado1x1
Salud: 3
Ejercito: 1
Fila:8
Columna: 7
```

```
Nombre: Soldado2x6
Salud: 5
Ejercito: 2
Fila:9
Columna: 3
Nombre: Soldado2x4
Salud: 1
Ejercito: 2
Fila:10
Columna: 1
Nombre: Soldado1x3
Salud: 1
Ejercito: 1
Fila:10
Columna: 2
Nombre: Soldado1x6
Salud: 3
Ejercito: 1
Fila:10
Columna: 8
Nombre: Soldado2x2
Salud: 3
Ejercito: 2
Fila:10
Columna: 9
```

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

El soldado con mas vida del ejercito 1 es:

Nombre: Soldado1x4

Vida: 3

Fila: 1

Fila: 1

Columna: 3

Ejercito1

El soldado con mas vida del ejercito 2 es:

Nombre: Soldado2x7

Vida: 5

Fila: 7

Fila: 7

Columna: 3

Ejercito2

Promedio de Vida del Ejercito 1: 2

Promedio de Vida del Ejercito 2: 3

Vida de todos los soldados:

Vida de Soldado2x8: 1

Vida de Soldado2x4: 1

Vida de Soldado2x1: 2

Vida de Soldado1x3: 1

Vida de Soldado1x4: 3

Vida de Soldado2x7: 5

Vida de Soldado2x6: 5

Vida de Soldado2x3: 4

Vida de Soldado1x5: 1

Vida de Soldado1x1: 3

Vida de Soldado1x2: 2

Vida de Soldado1x6: 3

Vida de Soldado2x5: 5

Vida de Soldado2x2: 3


```
Ranking de Vida segun Ordenamiento Burbuja
Nombre: Soldado2x5
Vida: 5
Fila: 6
Columna9
Ejercito2
Nombre: Soldado2x6
Vida: 5
Fila: 9
Columna3
Ejercito2
Nombre: Soldado2x7
Vida: 5
Fila: 7
Columna3
Ejercito2
Nombre: Soldado2x3
Vida: 4
Fila: 8
Columna4
Ejercito2
Nombre: Soldado2x2
Vida: 3
Fila: 10
Columna9
Ejercito2
Nombre: Soldado1x6
Vida: 3
Fila: 10
Columna8
Ejercito1
Nombre: Soldado1x1
Vida: 3
Fila: 8
Columna7
Ejercito1
Nombre: Soldado1x4
Vida: 3
Fila: 1
Columna3
Ejercito1
Nombre: Soldado1x2
Vida: 2
Fila: 4
Columna8
```

```
Ejercito1
Nombre: Soldado2x1
Vida: 2
Fila: 8
Columna2
Ejercito2
Nombre: Soldado1x5
Vida: 1
Fila: 5
Columna7
Ejercito1
Nombre: Soldado1x3
Vida: 1
Fila: 10
Columna2
Ejercito1
Nombre: Soldado2x4
Vida: 1
Fila: 10
Columna1
Ejercito2
Nombre: Soldado2x8
Vida: 1
Fila: 5
Columna1
Ejercito2
-----
```

Ranking de Vida segun Ordenamiento por Insercion

Nombre: Soldado2x5

Vida: 5

Fila: 6

Columna9

Ejercito2

Nombre: Soldado2x6

Vida: 5

Fila: 9

Columna3

Ejercito2

Nombre: Soldado2x7

Vida: 5

Fila: 7

Columna3

Ejercito2

Nombre: Soldado2x3

Vida: 4

Fila: 8

Columna4

Ejercito2

Nombre: Soldado2x2

Vida: 3

Fila: 10

Columna9

Ejercito2

Nombre: Soldado1x6

Vida: 3

Fila: 10

Columna8

Ejercito1

Nombre: Soldado1x1

Vida: 3

Fila: 8

Columna7

Ejercito1

Nombre: Soldado1x4

Vida: 3

Fila: 1

Columna3

Ejercito1

Nombre: Soldado1x2

Vida: 2

Fila: 4

Columna8

```
Ejercito1
Nombre: Soldado2x1
Vida: 2
Fila: 8
Columna2
Ejercito2
Nombre: Soldado1x5
Vida: 1
Fila: 5
Columna7
Ejercito1
Nombre: Soldado1x3
Vida: 1
Fila: 10
Columna2
Ejercito1
Nombre: Soldado2x4
Vida: 1
Fila: 10
Columna1
Ejercito2
Nombre: Soldado2x8
Vida: 1
Fila: 5
Columna1
Ejercito2
```

```
-----
Jugador 1. Ingrese el soldado a mover(Soldado1x1) del Ejercito 1
2
Soldado no encontrado, por favor ingrese otro
soldado1x4
Soldado no encontrado, por favor ingrese otro
Soldado1x4
Ahora ingrese la direccion (w:up, s:down, d:rigth, a:left)
s
= = = = =
= = 1 = = = = =
= = = = =
= = = = = 1 = =
2 = = = = 1 = = =
= = = = = 2 = =
= = 2 = = = = =
= 2 = 2 = = 1 = = =
= = 2 = = = = =
2 1 = = = = 1 2 =
```

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>