


	<b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b> <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b> <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b>	
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
<b>Aprobación:</b> 2022/03/01	<b>Código:</b> GUIA-PRLE-001	<b>Página:</b> 1

## INFORME DE LABORATORIO

### (formato estudiante)

INFORMACIÓN BÁSICA					
<b>ASIGNATURA:</b>	<i>Fundamentos de la programación 2</i>				
<b>TÍTULO DE LA PRÁCTICA:</b>	<i>Definición de Clases de Usuario</i> <i>Clase Soldado</i>				
<b>NÚMERO DE PRÁCTICA:</b>	<i>10</i>	<b>AÑO LECTIVO:</b>	<i>2023</i>	<b>NRO. SEMESTRE:</b>	<i>2do Semestre</i>
<b>FECHA DE PRESENTACIÓN</b>	<i>25/12/2023</i>	<b>HORA DE PRESENTACIÓN</b>	<i>22/30/00</i>		
<b>INTEGRANTE (s)</b> <i>Juan Diego Gutiérrez Ccama</i>				<b>NOTA (0-20)</b>	<i>Nota colocada por el docente</i>
<b>DOCENTE(s):</b> <i>Linno Jose Pinto Oppe</i>					

RESULTADOS Y PRUEBAS
<b>I. EJERCICIOS RESUELTOS:</b>

```
*Soldado.java x Videojuego.java
1
2 public class Soldado {
3     private String nombre;
4     private int nivelAtaque;
5     private int nivelDefensa;
6     private int nivelVida;
7     private int velocidad;
8     private String actitud;
9     private int vidaActual;
10    private boolean vive;
11    private int posFila;
12    private char posCol;
13    private char figura;
14    public Soldado(String n) {
15        nombre = n;
16    }
17    public int atacar() {
18        actitud = "ofensiva";
19        return avanzar();
20    }
21    public void defender() {
22        velocidad = 0;
23        actitud = "defensiva";
24    }
25    public int avanzar() {
26        return velocidad+1;
27    }
28    public void retroceder() {
29        defender();
30        actitud = "defensiva";
31        velocidad = velocidad - 1;
32    }
33    public void serAtacado() {
34        nivelVida = nivelVida - 1;
35    }
36    public void huir() {
37        actitud = "fuga";
38    }
39    public void morir() {
40        actitud = "muerto";
41        nivelVida = 0;
42    }
43    //sets
```

## II. PRUEB

```

43 //sets
44 public void setVidaActual(int a) {
45     nivelVida = a;
46 }
47 public void setFila(int n){
48     posFila = n;
49 }
50 public void setCol(char n){
51     posCol = n;
52 }
53 //tres constructores sobrecargados
54 public Soldado(String nom, char fig){
55     nombre = nom;
56     nivelAtaque = (int)(Math.random() * 5 + 1);
57     nivelDefensa = (int)(Math.random() * 5 + 1);
58     nivelVida = (int)(Math.random() * 5 + 1);
59     vidaActual = nivelVida;
60     velocidad = 0;
61     actitud = "Defensiva";
62     vive = true;
63     posFila = (int)(Math.random() * 10 + 1);
64     posCol = numCol();
65     figura = fig;
66 }
67 public Soldado(String nom, int nivAtaq, int nivDef, int vid, char fig){
68     nombre = nom;
69     nivelAtaque = nivAtaq;
70     nivelDefensa = nivDef;
71     nivelVida = vid;
72     vidaActual = vid;
73     velocidad = 0;
74     actitud = "Defensiva";
75     vive = true;
76     posFila = (int)(Math.random() * 10 + 1);
77     posCol = numCol();
78     figura = fig;
79 }
80 public Soldado(String nom, int nivAtaq, int nivDef, int nivVida, int nivAct, int vel,
81     String act, boolean vivir, int pFila, char pCol, char fig){
82     nombre = nom;
83     nivelAtaque = nivAtaq;
84     nivelDefensa = nivDef;
85     nivelVida = nivVida;
86     vidaActual = nivAct;
87     velocidad = vel;
88     actitud = act;
89     vive = vivir;
90     posFila = pFila;
91     posCol = pCol;
92     figura = fig;
93 }

```

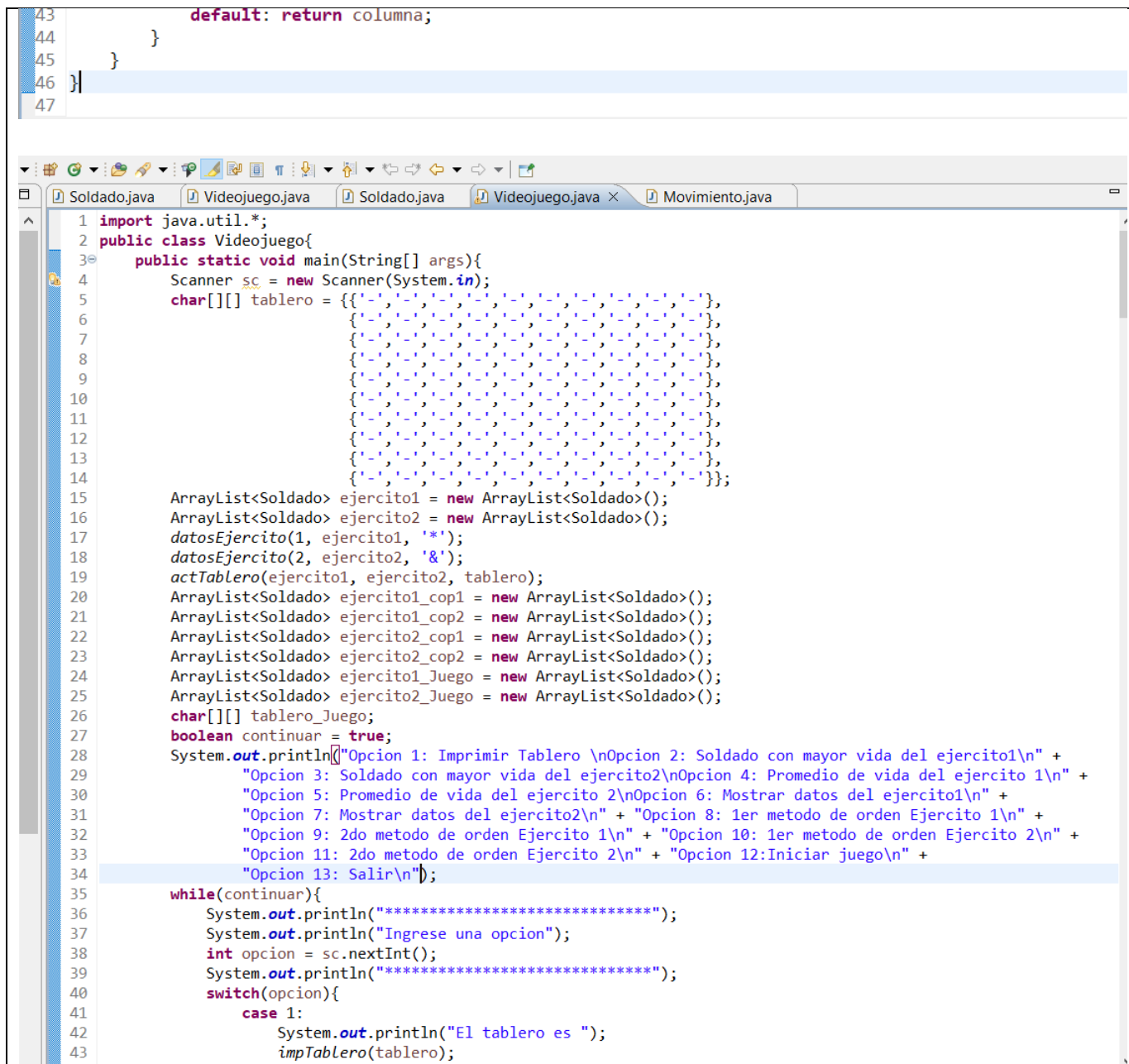
```
*Soldado.java x Videojuego.java
85     nivelVida = nivVida;
86     vidaActual = nivAct;
87     velocidad = vel;
88     actitud = act;
89     vive = vivir;
90     posFila = pFila;
91     posCol = pCol;
92     figura = fig;
93 }
94 public static char numCol() {
95     String a = "abcdefghij";
96     int n = (int)(Math.random() * a.length());
97     char car = a.charAt(n);
98     return car;
99 }
100 public String toString(){
101     return "Nombre: " + nombre + " Vida:" + vidaActual + " Fila:" + posFila +
102           " Columna:" + posCol + " Actitud:" + actitud;
103 }
104 //gets
105 public String getNombre(){
106     return nombre;
107 }
108 public int getAtaque(){
109     return nivelAtaque;
110 }
111 public int getDefensa(){
112     return nivelDefensa;
113 }
114 public int getNivVidAct(){
115     return vidaActual;
116 }
117 public int getVida(){
118     return nivelVida;
119 }
120 public int getVelocidad(){
121     return velocidad;
122 }
123 public String getActitud(){
124     return actitud;
125 }
126 public boolean getVive(){
127     return vive;
128 }
129 public int getFila(){
130     return posFila;
131 }
132 public char getColumna(){
133     return posCol;
134 }
135 public char getFigura(){
136     return figura;
137 }
138 }
```

```
1 public class Movimiento {
2     public static int movFila(int fila, int mov){
3         switch(mov){
4             case 1: return fila -= 1;
5             case 2: return fila -= 1;
6             case 3: return fila -= 1;
7             case 5: return fila += 1;
8             case 6: return fila += 1;
9             case 7: return fila += 1;
10            default: return fila;
11        }
12    }
13    public static int restFila(int fila, int mov){
14        switch(mov){
15            case 1: return fila += 1;
16            case 2: return fila += 1;
17            case 3: return fila += 1;
18            case 5: return fila -= 1;
19            case 6: return fila -= 1;
20            case 7: return fila -= 1;
21            default: return fila;
22        }
23    }
24    public static int movColumna(int columna, int mov){
25        switch(mov){
26            case 3: return columna += 1;
27            case 4: return columna += 1;
28            case 5: return columna += 1;
29            case 1: return columna -= 1;
30            case 7: return columna -= 1;
31            case 8: return columna -= 1;
32            default: return columna;
33        }
34    }
35    public static int restColumna(int columna, int mov){
36        switch(mov){
37            case 3: return columna -= 1;
38            case 4: return columna -= 1;
39            case 5: return columna -= 1;
40            case 1: return columna += 1;
41            case 7: return columna += 1;
42            case 8: return columna += 1;
43            default: return columna;
44        }
45    }
46 }
```

```

43         default: return columna;
44     }
45 }
46 }
47

```



```

1 import java.util.*;
2 public class Videojuego{
3     public static void main(String[] args){
4         Scanner sc = new Scanner(System.in);
5         char[][] tablero = {{' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
6                             {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
7                             {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
8                             {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
9                             {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
10                            {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
11                            {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
12                            {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
13                            {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '},
14                            {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '}};
15         ArrayList<Soldado> ejercito1 = new ArrayList<Soldado>();
16         ArrayList<Soldado> ejercito2 = new ArrayList<Soldado>();
17         datosEjercito(1, ejercito1, '*');
18         datosEjercito(2, ejercito2, '&');
19         actTablero(ejercito1, ejercito2, tablero);
20         ArrayList<Soldado> ejercito1_cop1 = new ArrayList<Soldado>();
21         ArrayList<Soldado> ejercito1_cop2 = new ArrayList<Soldado>();
22         ArrayList<Soldado> ejercito2_cop1 = new ArrayList<Soldado>();
23         ArrayList<Soldado> ejercito2_cop2 = new ArrayList<Soldado>();
24         ArrayList<Soldado> ejercito1_Juego = new ArrayList<Soldado>();
25         ArrayList<Soldado> ejercito2_Juego = new ArrayList<Soldado>();
26         char[][] tablero_Juego;
27         boolean continuar = true;
28         System.out.println("Opcion 1: Imprimir Tablero \nOpcion 2: Soldado con mayor vida del ejercito1\n" +
29                             "Opcion 3: Soldado con mayor vida del ejercito2\nOpcion 4: Promedio de vida del ejercito 1\n" +
30                             "Opcion 5: Promedio de vida del ejercito 2\nOpcion 6: Mostrar datos del ejercito1\n" +
31                             "Opcion 7: Mostrar datos del ejercito2\n" + "Opcion 8: 1er metodo de orden Ejercito 1\n" +
32                             "Opcion 9: 2do metodo de orden Ejercito 1\n" + "Opcion 10: 1er metodo de orden Ejercito 2\n" +
33                             "Opcion 11: 2do metodo de orden Ejercito 2\n" + "Opcion 12:Iniciar juego\n" +
34                             "Opcion 13: Salir\n");
35         while(continuar){
36             System.out.println("*****");
37             System.out.println("Ingrese una opcion");
38             int opcion = sc.nextInt();
39             System.out.println("*****");
40             switch(opcion){
41                 case 1:
42                     System.out.println("El tablero es ");
43                     impTablero(tablero);

```

```

44         break;
45     case 2:
46         mayorVida(ejercito1, 1);
47         break;
48     case 3:
49         mayorVida(ejercito2, 2);
50         break;
51     case 4:
52         promVida(ejercito1, 1);
53         break;
54     case 5:
55         promVida(ejercito2, 2);
56         break;
57     case 6:
58         System.out.println("Los datos del ejercito 1 son : ");

```

```

Soldado.java  Videojuego.java  Soldado.java  Videojuego.java x  Movimiento.java
58         System.out.println("Los datos del ejercito 1 son : ");
59         mostDatos(ejercito1);
60         break;
61     case 7:
62         System.out.println("Los datos del ejercito 2 son : ");
63         mostDatos(ejercito2);
64         break;
65     case 8:
66         System.out.println("Primer metodo de orden - Ejercito 1");
67         copiaDatos(ejercito1, ejercito1_cop1);
68         ordenarBurbuja(ejercito1_cop1);
69         ejercito1_cop1.clear();
70         break;
71     case 9:
72         System.out.println("2do metodo de orden - Ejercito 1");
73         copiaDatos(ejercito1, ejercito1_cop2);
74         ordenarSeleccion(ejercito1_cop2);
75         ejercito1_cop2.clear();
76         break;
77     case 10:
78         System.out.println("1er metodo de orden - Ejercito 2");
79         copiaDatos(ejercito2, ejercito2_cop1);
80         ordenarBurbuja(ejercito2_cop1);
81         ejercito2_cop1.clear();
82         break;
83     case 11:
84         System.out.println("2do metodo de orden - Ejercito 2");
85         copiaDatos(ejercito2, ejercito2_cop2);
86         ordenarSeleccion(ejercito2_cop2);
87         ejercito2_cop2.clear();
88         break;
89     case 12:
90         System.out.println("Inicio de Juego");
91         tablero_Juego = tableroCopia(tablero);
92         copiaDatos(ejercito1, ejercito1_Juego);
93         copiaDatos(ejercito2, ejercito2_Juego);
94         juego(ejercito1_Juego, ejercito2_Juego, tablero_Juego);
95         ejercito1_Juego.clear();
96         ejercito2_Juego.clear();
97     case 13:
98         System.out.println("Saliendo del programa");
99         continuar = false;
100        break;

```

```

Soldado.java  Videojuego.java  Soldado.java  Videojuego.java x  Movimiento.java
101         default:
102             System.out.println("Opcion no valida");
103         }
104     }
105 }
106 public static void juego(ArrayList<Soldado> ejercito1, ArrayList<Soldado> ejercito2, char[][] tablero){
107     boolean validez = true;
108     String coordenada;
109     int movimiento, fila, columna;
110     int i = 0;
111     while(validez){
112         if(ejercito1.size() == 0 || ejercito2.size() == 0){
113             validez = false;
114             break;
115         } else if(i % 2 == 0){
116             System.out.println("Turno del Ejercito 1");
117             coordenada = ingresar(ejercito1.get(0).getFigura(), tablero);
118             movimiento = ingresarMovimiento(coordenada);
119             movimientoJugado(coordenada, movimiento, ejercito1, ejercito2, tablero);
120             impTablero(tablero);
121         } else {
122             System.out.println("Turno del Ejercito 2");
123             coordenada = ingresar(ejercito2.get(0).getFigura(), tablero);
124             movimiento = ingresarMovimiento(coordenada);
125             movimientoJugado(coordenada, movimiento, ejercito2, ejercito1, tablero);
126             impTablero(tablero);
127         }
128         i++;
129     }
130     if(ejercito1.size() == 0){
131         System.out.println("Salio victorio el ejercito 2");
132         for(Soldado n: ejercito2){
133             System.out.println(n);
134         }
135     } else{
136         System.out.println("Salio victorioso el ejercito 1");
137         for(Soldado m: ejercito2){
138             System.out.println(m);
139         }
140     }
141 }
142 public static void movimientoJugado(String coordenada, int movimiento, ArrayList<Soldado> usuario, ArrayList<Sc
143     int fila_act, columna_act, fila_mov, columna_mov, posAct;

```



```

143     int fila_act, columna_act, fila_mov, columna_mov, posAct;
144     fila_act = conversionFila(coordenada.substring(1, coordenada.length())) - 1;
145     columna_act = nroColumna(coordenada.charAt(0));
146     fila_mov = Movimiento.movFila(fila_act, movimiento);
147     columna_mov = Movimiento.movColumna(columna_act, movimiento);
148     if(tablero[fila_mov][columna_mov] == usuario.get(0).getFigura()){
149         System.out.println("Posicion ocupada por una figura de tu mismo ejercito");
150     } else if(tablero[fila_mov][columna_mov] == '-'){
151         posAct = buscadorPosicion(usuario, coordenada);
152         usuario.get(posAct).setFila(fila_mov + 1);
153         usuario.get(posAct).setCol(conversionBusqueda(columna_mov));
154         tablero[fila_act][columna_act] = '-';
155         tablero[fila_mov][columna_mov] = usuario.get(0).getFigura();
156     } else {
157         System.out.println("Enemigo hallado");
158         batalla(coordenada, movimiento, usuario, contrincante, tablero);
159     }
160 }
161
162 public static void batalla(String coordenada, int movimiento, ArrayList<Soldado> usuario, ArrayList<Soldado> co
163     int fila_act, columna_act, fila_mov, columna_mov, posAct, us, cont;
164     fila_act = conversionFila(coordenada.substring(1, coordenada.length())) - 1;
165     columna_act = nroColumna(coordenada.charAt(0));
166     fila_mov = Movimiento.movFila(fila_act, movimiento);
167     columna_mov = Movimiento.movColumna(columna_act, movimiento);
168     String movRealizado = conversionBusqueda(columna_mov) + String.valueOf(fila_mov + 1);
169     us = buscadorPosicion(usuario, coordenada);
170     usuario.get(us).atacar();
171     cont = buscadorPosicion(contrincante, movRealizado);
172     contrincante.get(cont).defender();
173     System.out.println("Jugador actual : " + usuario.get(us).getNivVidAct());
174     System.out.println("Jugador enemigo : " + contrincante.get(cont).getNivVidAct());
175     if(usuario.get(us).getNivVidAct() == contrincante.get(cont).getNivVidAct()){
176         int suerte = (int)(Math.random() * 2);
177         if(suerte == 0){
178             contrincante.remove(cont);
179             usuario.get(us).setFila(fila_mov + 1);
180             usuario.get(us).setCol(conversionBusqueda(columna_mov));
181             tablero[fila_mov][columna_mov] = usuario.get(us).getFigura();
182             tablero[fila_act][columna_act] = '-';
183         } else {
184             usuario.remove(us);
185             tablero[fila_act][columna_act] = '-';
186         }
187     }
188     } else if(usuario.get(us).getNivVidAct() > contrincante.get(cont).getNivVidAct()){
189         contrincante.remove(cont);
190         usuario.get(us).setFila(fila_mov + 1);
191         usuario.get(us).setCol(conversionBusqueda(columna_mov));
192         tablero[fila_mov][columna_mov] = usuario.get(us).getFigura();
193         tablero[fila_act][columna_act] = '-';
194     } else {
195         usuario.remove(us);
196         tablero[fila_act][columna_act] = '-';
197     }
198 }
199
200 public static String ingresar(char n, char[][] tablero){
201     String coordenada = "";
202     int fila, columna;
203     boolean validez = true;
204     Scanner sc = new Scanner(System.in);
205     while(validez){
206         System.out.println("1 2 3\n8 x 4\n7 6 5");
207         System.out.println("Ingresa otra coordenada");
208         coordenada = sc.next();
209         fila = conversionFila(coordenada.substring(1, coordenada.length()));

```

```

Soldado.java  Videojuego.java  Soldado.java  Videojuego.java x  Movimiento.java
206     fila = conversionFila(coordenada.substring(1, coordenada.length()));
207     columna = nroColumna(coordenada.charAt(0));
208     System.out.println(fila + " " + columna);
209     if(tablero[fila - 1][columna] == n){
210         break;
211     } else {
212         System.out.println("Coordenada incorrecta");
213     }
214 }
215 return coordenada;
216 }
217 public static int ingresarMovimiento(String coordenada){
218     int movimiento = 0;
219     boolean validez = true;
220     int fila, columna;
221     Scanner sc = new Scanner(System.in);
222     while(validez){
223         System.out.println("Ingresar Movimiento: ");
224         movimiento = sc.nextInt();
225         fila = conversionFila(coordenada.substring(1, coordenada.length())) - 1;
226         columna = nroColumna(coordenada.charAt(0));
227         fila = Movimiento.movFila(fila, movimiento);
228         columna = Movimiento.movColumna(columna, movimiento);
229         System.out.println(fila + " Movimiento aplicado " + columna);
230         if(validezMovimiento(fila, columna)){
231             break;
232         } else {
233             System.out.println("Movimiento invalido");
234             fila = Movimiento.restFila(fila, movimiento);
235             columna = Movimiento.restColumna(columna, movimiento);
236         }
237     }
238     return movimiento;
239 }
240 public static boolean validezMovimiento(int fila, int columna){
241     return fila <= 9 && fila >= 0 && columna <= 9 && columna >= 0;
242 }
243 public static void datosEjercito(int n, ArrayList<Soldado> ejercito, char fig){
244     int nroSoldados = (int)(Math.random() * 10 + 1);
245     System.out.println("El ejercito " + n + " tiene un total de " + nroSoldados + " soldados");
246     for(int i = 0; i < nroSoldados; i++){
247         String nombre = "Soldado " + (i + 1) + "X" + n;
248         ejercito.add(new Soldado(nombre, fig));
249     }
250 }
251 public static void actTablero(ArrayList<Soldado> ejercito1, ArrayList<Soldado> ejercito2, char[][] tablero){
252     int fila;
253     char columna;
254     for(int i = 0; i < ejercito1.size(); i++){
255         fila = ejercito1.get(i).getFila();
256         columna = ejercito1.get(i).getColumna();
257         while(tablero[fila - 1][nroColumna(columna)] != '-'){
258             fila = (int)(Math.random() * 10 + 1);
259             columna = Soldado.numCol();
260         }
261         ejercito1.get(i).setFila(fila);
262         ejercito1.get(i).setCol(columna);
263         tablero[fila - 1][nroColumna(columna)] = ejercito1.get(i).getFigura();
264     }

```

```

265     for(int j = 0; j < ejercito2.size(); j++){
266         fila = ejercito2.get(j).getFila();
267         columna = ejercito2.get(j).getColumna();
268         while(tablero[fila - 1][nroColumna(columna)] != '-'){
269             fila = (int)(Math.random() * 10 + 1);
270             columna = Soldado.numCol();
271         }
272         ejercito2.get(j).setFila(fila);
273         ejercito2.get(j).setCol(columna);
274         tablero[fila - 1][nroColumna(columna)] = ejercito2.get(j).getFigura();
275     }
276 }
277 public static void impTablero(char[][] tablero){
278     System.out.println("El tablero es :");
279     for(int x = 0; x < tablero.length; x++){
280         for(int y = 0; y < tablero[x].length; y++){
281             System.out.print(tablero[x][y]);
282         }
283         System.out.println();
284     }
285 }
286 public static int nroColumna(char n){
287     switch(n){
288         case 'A': return 0;
289         case 'B': return 1;
290         case 'C': return 2;
291         case 'D': return 3;
292         case 'E': return 4;
293         case 'F': return 5;
294         case 'G': return 6;
295         case 'H': return 7;
296         case 'I': return 8;
297         case 'J': return 9;
298         default: return 0;
299     }
300 }
301 public static int conversionFila(String n){
302     switch(n){
303         case "1": return 1;
304         case "2": return 2;
305         case "3": return 3;
306         case "4": return 4;
307         case "5": return 5;
308         case "6": return 6;
309         case "7": return 7;
310         case "8": return 8;
311         case "9": return 9;
312         case "10": return 10;
313         default: return 0;
314     }
315 }
316 public static char conversionBusqueda(int m){
317     switch(m){
318         case 0: return 'A';
319         case 1: return 'B';
320         case 2: return 'C';
321         case 3: return 'D';
322         case 4: return 'E';
323         case 5: return 'F';
324         case 6: return 'G';
325         case 7: return 'H';
326         case 8: return 'I';
327         case 9: return 'J';
328         default: return ' ';
329     }
330 }

```

```

324         case 6: return 'G';
325         case 7: return 'H';
326         case 8: return 'I';
327         case 9: return 'J';
328         default: return 0;
329     }
330 }
331 public static char[][] tableroCopia(char[][] tablero){
332     char[][] tableroNuevo = new char[tablero.length][tablero[0].length];
333     for(int i = 0; i < tablero.length; i++){
334         for(int j = 0; j < tablero[0].length; j++){
335             tableroNuevo[i][j] = tablero[i][j];
336         }
337     }
338     return tableroNuevo;
339 }
340 public static int buscadorPosicion(ArrayList<Soldado> ejercito, String posicion){
341     for(int i = 0; i < ejercito.size(); i++){
342         if(posicion.charAt(0) == ejercito.get(i).getColumna() && conversionFila(posicion.substring(1, posicion.
343             return i;
344         }
345     }
346     return 0;
347 }
348 public static void mayorVida(ArrayList<Soldado> ejercito, int n){
349     int mayor = ejercito.get(0).getVida();
350     for(int i = 1; i < ejercito.size(); i++){
351         if(ejercito.get(i).getVida() > mayor)
352             mayor = ejercito.get(i).getVida();
353     }
354     System.out.println("Los soldados con mayor vida del ejercito " + n + " son:");
355     for(int j = 0; j < ejercito.size(); j++){
356         if(ejercito.get(j).getVida() == mayor){
357             System.out.println(ejercito.get(j));
358         }
359     }
360 }
361 public static void promVida(ArrayList<Soldado> ejercito, int m){
362     double n = 0;
363     for(int i = 0; i < ejercito.size(); i++){
364         n += ejercito.get(i).getVida();
365     }
366     n /= ejercito.size();
367     System.out.println("El promedio de vida del ejercito " + m + " es " + n);

```

	<p align="center"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
<b>Aprobación:</b> 2022/03/01	<b>Código:</b> GUIA-PRLE-001	<b>Página:</b> 13

```

366      System.out.println("El promedio de vida del ejercito es " + n);
367  }
368  public static void copiaDatos(ArrayList<Soldado> ejercito, ArrayList<Soldado> copia){
369      for(int i = 0; i < ejercito.size(); i++){
370          Soldado pos = ejercito.get(i);
371          Soldado c = new Soldado(pos.getNombre(), pos.getAtaque(), pos.getDefensa(), pos.getVida(), pos.getNivV:
372          copia.add(c);
373      }
374  }
375  public static void ordenarSeleccion(ArrayList<Soldado> ejercito){
376      for(int i = 0; i < ejercito.size() - 1; i++){
377          for(int j = i + 1; j < ejercito.size(); j++){
378              Soldado may = ejercito.get(i);
379              Soldado men = ejercito.get(j);
380              if(men.getVida() > may.getVida()){
381                  Soldado mayor = new Soldado(men.getNombre(), men.getAtaque(), men.getDefensa(), men.getVida(),
382                  Soldado menor = new Soldado(may.getNombre(), may.getAtaque(), may.getDefensa(), may.getVida(),
383                  ejercito.set(i, mayor);
384                  ejercito.set(j, menor);
385              }
386          }
387      }
388      mostDatos(ejercito);
389  }
390  public static void ordenarBurbuja(ArrayList<Soldado> ejercito){
391      for(int i = 0; i < ejercito.size() - 1; i++){
392          for(int j = 0; j < ejercito.size() - 1; j++){
393              Soldado may = ejercito.get(i);
394              Soldado men = ejercito.get(i + 1);
395              if(men.getVida() > may.getVida()){
396                  Soldado mayor = new Soldado(men.getNombre(), men.getAtaque(), men.getDefensa(), men.getVida(),
397                  Soldado menor = new Soldado(may.getNombre(), may.getAtaque(), may.getDefensa(), may.getVida(),
398                  ejercito.set(j, mayor);
399                  ejercito.set(j + 1, menor);
400              }
401          }
402      }
403      mostDatos(ejercito);
404  }
405  public static void mostDatos(ArrayList<Soldado> ejercito){
406      for(Soldado n: ejercito)
407          System.out.println(n);
408  }
409  }

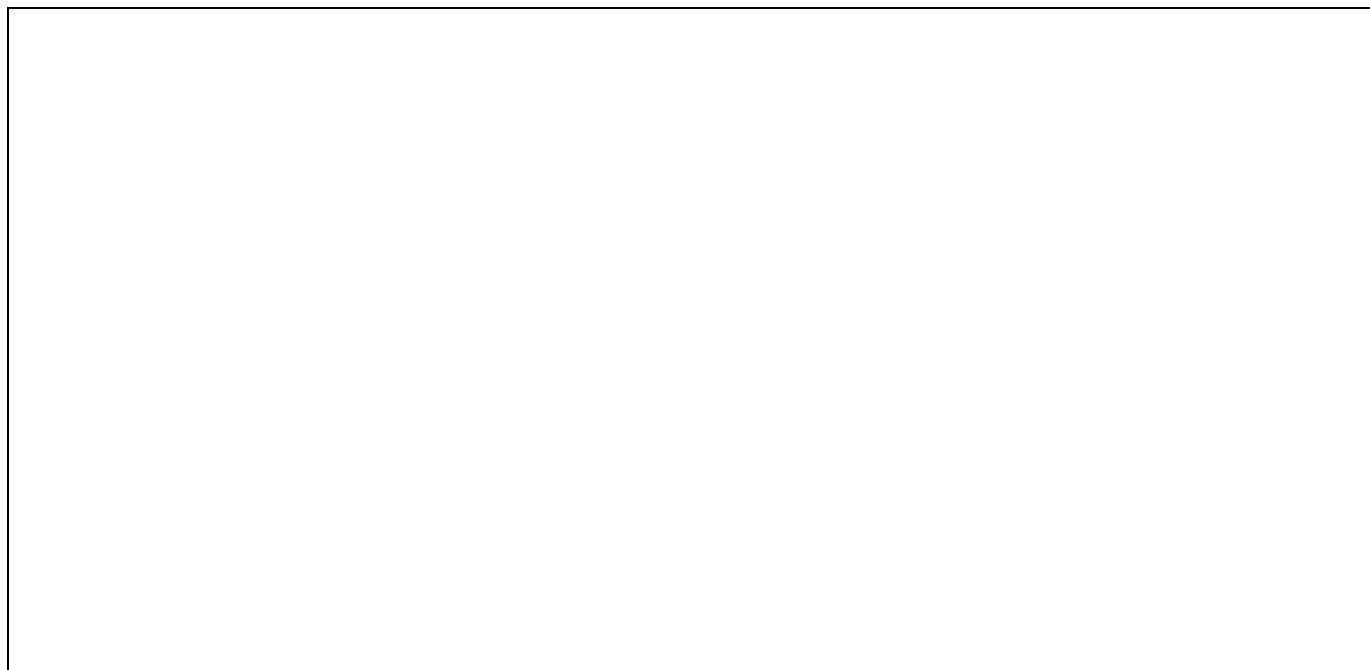
```

	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 14

```
<terminated> Videojuego (1) [Java Application] C:\Users\UJKM
El ejercito 1 tiene un total de 6 soldados
El ejercito 2 tiene un total de 3 soldados
Opcion 1: Imprimir Tablero
Opcion 2: Soldado con mayor vida del ejercito1
Opcion 3: Soldado con mayor vida del ejercito2
Opcion 4: Promedio de vida del ejercito 1
Opcion 5: Promedio de vida del ejercito 2
Opcion 6: Mostrar datos del ejercito1
Opcion 7: Mostrar datos del ejercito2
Opcion 8: 1er metodo de orden Ejercito 1
Opcion 9: 2do metodo de orden Ejercito 1
Opcion 10: 1er metodo de orden Ejercito 2
Opcion 11: 2do metodo de orden Ejercito 2
Opcion 12:Iniciar juego
Opcion 13: Salir

*****
Ingrese una opcion
1
*****
El tablero es
El tablero es :
&-----
*-----
-----
&-----
*-----
&-----
*-----
*-----
*-----
*-----
*****
Ingrese una opcion
12
*****
Inicio de Juego
Turno del Ejercito 1
1 2 3
8 x 4
7 6 5
```

	<p align="center"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
<b>Aprobación:</b> 2022/03/01	<b>Código:</b> GUIA-PRLE-001	<b>Página:</b> 15



### III. CUESTIONARIO:

REPOSITORIO: <https://github.com/UJKMjuandi/FundamentosP2>

## CONCLUSIONES

La expansión del programa para permitir un enfrentamiento entre jugadores humanos ha llevado a una mejora significativa en la experiencia del usuario y la dinámica del juego. La implementación de nuevos métodos específicos para la interacción humana ha enriquecido la funcionalidad del sistema y ha añadido una capa estratégica al enfrentamiento.

En primer lugar, la introducción de métodos que permiten a los jugadores humanos tomar decisiones estratégicas, como la selección de soldados y la planificación de movimientos, ha elevado la interactividad del juego. Este enfoque ofrece a los jugadores la oportunidad de participar activamente en el desarrollo del juego, tomando decisiones tácticas que afectan directamente al resultado de la batalla.

	<p align="center"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
<b>Aprobación:</b> 2022/03/01	<b>Código:</b> GUIA-PRLE-001	<b>Página:</b> 16

La gestión de turnos y la dinámica del juego también ha sido mejorada mediante la implementación de métodos que permiten una alternancia justa de acciones entre los jugadores. Este enfoque equitativo asegura una experiencia de juego balanceada y brinda a ambos participantes la oportunidad de desplegar estrategias y tácticas.

La interfaz de usuario se ha visto mejorada significativamente con la incorporación de métodos que facilitan la presentación clara de información relevante y la interacción fluida durante el juego. Esto incluye la visualización detallada del tablero, la selección intuitiva de soldados y la toma de decisiones estratégicas con una interfaz fácil de usar.

La implementación de métodos para manejar eventos y resultados durante el enfrentamiento humano vs humano garantiza una evaluación precisa de las acciones, cálculos precisos de resultados y una actualización coherente del estado del juego. Este enfoque contribuye a una experiencia de juego sin problemas y libre de errores.

Considerando las complejidades estratégicas, la implementación de métodos para evaluar fortalezas y debilidades de los soldados, así como la planificación y ejecución de estrategias tácticas, brinda a los jugadores la oportunidad de participar en una experiencia de juego más rica y desafiante.

### METODOLOGÍA DE TRABAJO

- 1.- *Leer los enunciados cuidadosamente para saber que requiere el problemas.*
- 2.- *verificar el diagrama uml puesto en la practica*
- 3.- *Empezar a hacer las clase soldado con el uml*
- 4.- *Realizar el Videojuego con los métodos apropiados.*


### REFERENCIAS Y BIBLIOGRAFÍA

[1] M. A. Lopez, E. Castro Gutierrez, *Fundamentos de la Programación 2 Topics de Programación Orientada a Objetos*. Arequipa: UNSA, 2021

### RUBRICA PARA EL CONTENIDO DEL INFORME Y DEMOSTRACIÓN

El alumno debe marcar o dejar en blanco en celdas de la columna Checklist si cumplió con el ítem correspondiente.



	<b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b> <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b> <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b>	
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
<b>Aprobación:</b> 2022/03/01	<b>Código:</b> GUIA-PRLE-001	<b>Página:</b> 17

Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.



El alumno debe autocalificarse en la columna Estudiante de acuerdo a la siguiente tabla:

Tabla 1: Niveles de desempeño

Nivel				
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	3	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	0	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	3	
TOTAL		20		14	

Tabla 2: Rúbrica para contenido del Informe y demostración

 <b>INGENIERIA</b> <b>ISTEMAS</b>	<b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b> <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b> <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b>	 <b>ABET</b>
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
<b>Aprobación:</b> 2022/03/01	<b>Código:</b> GUIA-PRLE-001	<b>Página:</b> 18