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

INFORME DE LABORATORIO

(formato estudiante)

INFORMACIÓN BÁSICA					
ASIGNATURA:	<i>Fundamentos de la programación 2</i>				
TÍTULO DE LA PRÁCTICA:	<i>Definición de Clases de Usuario</i> <i>Clase Soldado</i>				
NÚMERO DE PRÁCTICA:	<i>11</i>	AÑO LECTIVO:	<i>2023</i>	NRO. SEMESTRE:	<i>2do Semestre</i>
FECHA DE PRESENTACIÓN	<i>26/12/2023</i>	HORA DE PRESENTACIÓN	<i>15/45/00</i>		
INTEGRANTE (s) <i>Juan Diego Gutiérrez Ccama</i>				NOTA (0-20)	<i>Nota colocada por el docente</i>
DOCENTE(s): <i>Linno Jose Pinto Oppe</i>					

RESULTADOS Y PRUEBAS
I. EJERCICIOS RESUELTOS:

```
*Soldado.java × 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
```

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 3</p>



II. PRUEB

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 4</p>

```

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 }

```

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 5</p>

```

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 public void addVida(){
139     vidaActual++;
140 }
141 }
142 }
```

```
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.ArrayList;
2 import java.util.Scanner;
3 public class VideoJuego{
4     public static void main(String[] args){
5         Scanner sc = new Scanner(System.in);
6         char[][] tablero = {{
7             ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
8             ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
9             ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
10            ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
11            ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
12            ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
13            ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
14            ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
15            ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '
16        }};
17        ArrayList<Soldado> ejercito1 = new ArrayList<Soldado>();
18        ArrayList<Soldado> ejercito2 = new ArrayList<Soldado>();
19        datosEjercito(1, ejercito1, '*');
20        datosEjercito(2, ejercito2, '&');
21        actTablero(ejercito1, ejercito2, tablero);
22        ArrayList<Soldado> ejercito1_cop1 = new ArrayList<Soldado>();
23        ArrayList<Soldado> ejercito1_cop2 = new ArrayList<Soldado>();
24        ArrayList<Soldado> ejercito2_cop1 = new ArrayList<Soldado>();
25        ArrayList<Soldado> ejercito2_cop2 = new ArrayList<Soldado>();
26        ArrayList<Soldado> ejercito1_Juego = new ArrayList<Soldado>();
27        ArrayList<Soldado> ejercito2_Juego = new ArrayList<Soldado>();
28        char[][] tablero_Juego;
29        boolean continuar = true;
30        System.out.println("Opcion 1: Imprimir Tablero \nOpcion 2: Soldado con mayor vida del ejercito1\n" +
31            "Opcion 3: Soldado con mayor vida del ejercito2\nOpcion 4: Promedio de vida del ejercito 1\n" +
32            "Opcion 5: Promedio de vida del ejercito 2\nOpcion 6: Mostrar datos del ejercito1\n" +
33            "Opcion 7: Mostrar datos del ejercito2\n" + "Opcion 8: 1er metodo de orden Ejercito 1\n" +
34            "Opcion 9: 2do metodo de orden Ejercito 1\n" + "Opcion 10: 1er metodo de orden Ejercito 2\n" +
35            "Opcion 11: 2do metodo de orden Ejercito 2\n" + "Opcion 12: Iniciar juego\n" +
36            "Opcion 13: Salir\n");
37        while(continuar){
38            System.out.println("*****");
39            System.out.println("Ingrese una opcion");
40            int opcion = sc.nextInt();
41            System.out.println("*****");
42            switch(opcion){
43                case 1:
44                    System.out.println("El tablero es ");
45                    System.out.println(tablero);
46                    impTablero(tablero);
47                    break;
48                case 2:
49                    mayorVida(ejercito1, 1);
50                    break;
51                case 3:
52                    mayorVida(ejercito2, 2);
53                    break;
54                case 4:
55                    promVida(ejercito1, 1);
56                    break;
57                case 5:
58                    promVida(ejercito2, 2);
59                    break;
60            }
61        }
62    }
63 }

```


	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 9</p>

```

57         break;
58     case 6:
59         System.out.println("Los datos del ejercito 1 son : ");
60         mostDatos(ejercito1);
61         break;
62     case 7:
63         System.out.println("Los datos del ejercito 2 son : ");
64         mostDatos(ejercito2);
65         break;
66     case 8:
67         System.out.println("Primer metodo de orden - Ejercito 1");
68         copiaDatos(ejercito1, ejercito1_cop1);
69         ordenarBurbuja(ejercito1_cop1);
70         ejercito1_cop1.clear();
71         break;
72     case 9:
73         System.out.println("2do metodo de orden - Ejercito 1");
74         copiaDatos(ejercito1, ejercito1_cop2);
75         ordenarSeleccion(ejercito1_cop2);
76         ejercito1_cop2.clear();
77         break;
78     case 10:
79         System.out.println("1er metodo de orden - Ejercito 2");
80         copiaDatos(ejercito2, ejercito2_cop1);
81         ordenarBurbuja(ejercito2_cop1);
82         ejercito2_cop1.clear();
83         break;
84     case 11:
85         System.out.println("2do metodo de orden - Ejercito 2");
86         copiaDatos(ejercito2, ejercito2_cop2);
87         ordenarSeleccion(ejercito2_cop2);
88         ejercito2_cop2.clear();
89         break;
90     case 12:
91         System.out.println("Inicio de Juego");
92         tablero_Juego = tableroCopia(tablero);
93         copiaDatos(ejercito1, ejercito1_Juego);
94         copiaDatos(ejercito2, ejercito2_Juego);
95         impTablero(tablero_Juego);
96         juego(ejercito1_Juego, ejercito2_Juego, tablero_Juego);
97         ejercito1_Juego.clear();
98         ejercito2_Juego.clear();
99     case 13:
100         System.out.println("Saliendo del programa");
101         continuar = false;
102         break;
103     default:
104         System.out.println("Opcion no valida");
105 }
106 }
107 }
108 public static void juego(ArrayList<Soldado> ejercito1, ArrayList<Soldado> ejercito2, char[][] tablero){
109     boolean validez = true;
110     String coordenada;
111     int movimiento, fila, columna;
112     int i = 0;
113     while(validez){
114         if(ejercito1.size() == 0 || ejercito2.size() == 0){
115             validez = false;
116             break;
117         } else if(i % 2 == 0){
118             System.out.println("Turno del Ejercito 1");
119             coordenada = ingresar(ejercito1.get(0).getFigura(), tablero);
120             movimiento = ingresarMovimiento(coordenada);
121             movimientoJugado(coordenada, movimiento, ejercito1, ejercito2, tablero);

```

```

122         impTablero(tablero);
123     } else {
124         System.out.println("Turno del Ejercito 2");
125         coordenada = ingresar(ejercito2.get(0).getFigura(), tablero);
126         movimiento = ingresarMovimiento(coordenada);
127         movimientoJugado(coordenada, movimiento, ejercito2, ejercito1, tablero);
128         impTablero(tablero);
129     }
130     i++;
131 }
132 if(ejercito1.size() == 0){
133     System.out.println("Salio victorio el ejercito 2");
134     for(Soldado n: ejercito2){
135         System.out.println(n);
136     }
137 } else{
138     System.out.println("Salio victorioso el ejercito 1");
139     for(Soldado m: ejercito2){
140         System.out.println(m);
141     }
142 }
143 }
144 public static void movimientoJugado(String coordenada, int movimiento, ArrayList<Soldado> usuario, ArrayList<Soldado> contrincante,
145 int fila_act, columna_act, fila_mov, columna_mov, posAct;
146 fila_act = Integer.parseInt(coordenada.substring(1, coordenada.length())) - 1;
147 columna_act = nroColumna(coordenada.charAt(0));
148 fila_mov = Movimiento.movFila(fila_act, movimiento);
149 columna_mov = Movimiento.movColumna(columna_act, movimiento);
150 if(tablero[fila_mov][columna_mov] == usuario.get(0).getFigura()){
151     System.out.println("Posicion ocupada por una figura de tu mismo ejercito");
152 } else if(tablero[fila_mov][columna_mov] == '-'){
153     posAct = buscadorPosicion(usuario, coordenada);
154     usuario.get(posAct).setFila(fila_mov + 1);
155     usuario.get(posAct).setCol(conversionBusqueda(columna_mov));
156     tablero[fila_act][columna_act] = '-';
157     tablero[fila_mov][columna_mov] = usuario.get(0).getFigura();
158 } else {
159     System.out.println("Enemigo hallado");
160     batalla(coordenada, movimiento, usuario, contrincante, tablero);
161 }
162 }
163 public static void batalla(String coordenada, int movimiento, ArrayList<Soldado> usuario, ArrayList<Soldado> contrincante, char
164 int fila_act, columna_act, fila_mov, columna_mov, posAct, us, cont, vidaAd;
165 fila_act = Integer.parseInt(coordenada.substring(1, coordenada.length())) - 1;
166 columna_act = nroColumna(coordenada.charAt(0));
167 fila_mov = Movimiento.movFila(fila_act, movimiento);
168 columna_mov = Movimiento.movColumna(columna_act, movimiento);
169 String movRealizado = conversionBusqueda(columna_mov) + String.valueOf(fila_mov + 1);
170 us = buscadorPosicion(usuario, coordenada);
171 usuario.get(us).atacar();
172 cont = buscadorPosicion(contrincante, movRealizado);
173 contrincante.get(cont).defender();
174 System.out.println("Jugador actual : " + usuario.get(us).getNivVidAct());
175 System.out.println("Jugador enemigo : " + contrincante.get(cont).getNivVidAct());
176 int total = usuario.get(us).getNivVidAct() + contrincante.get(cont).getNivVidAct();
177 double probabilidad1 = usuario.get(us).getNivVidAct() * 100 / total;
178 probabilidad1 = Math.round(probabilidad1 * 10.0) / 10.0;
179 System.out.println("Posibilidad de porcentaje del Jugador actual: " + probabilidad1 + "%");
180 System.out.println("Posibilidad de porcentaje del Jugador enemigo: " + (100 - probabilidad1) + "%");
181 int nroRandom = (int)(Math.random() * 100 + 1);
182 System.out.println("El numero aleatorio fue de " + nroRandom);
183 if(0 <= nroRandom && nroRandom <= probabilidad1){
184     contrincante.remove(cont);

```

```



185     usuario.get(us).setFila(fila_mov + 1);
186     usuario.get(us).setCol(conversionBusqueda(columna_mov));
187     tablero[fila_mov][columna_mov] = usuario.get(us).getFigura();
188     tablero[fila_act][columna_act] = '-';
189     usuario.get(us).addVida();
190 } else {
191     usuario.remove(us);
192     tablero[fila_act][columna_act] = '-';
193     contrincante.get(cont).addVida();
194 }
195 }
196 public static String ingresar(char n, char[][] tablero){
197     String coordenada = "";
198     int fila, columna;
199     boolean validez = true;
200     Scanner sc = new Scanner(System.in);
201     while(validez){
202         System.out.println("1 2 3\n8 x 4\n7 6 5");
203         System.out.println("Ingresa otra coordenada");
204         coordenada = sc.next();
205         fila = Integer.parseInt(coordenada.substring(1, coordenada.length()));
206         columna = nroColumna(coordenada.charAt(0));
207         System.out.println(fila + " " + columna);
208         if(tablero[fila - 1][columna] == n){
209             break;
210         } else {
211             System.out.println("Coordenada incorrecta");
212         }
213     }
214     return coordenada;
215 }
216 public static int ingresarMovimiento(String coordenada){
217     int movimiento = 0;
218     boolean validez = true;
219     int fila, columna;
220     Scanner sc = new Scanner(System.in);
221     while(validez){
222         System.out.println("Ingresa Movimiento: ");
223         movimiento = sc.nextInt();
224         fila = Integer.parseInt(coordenada.substring(1, coordenada.length())) - 1;
225         columna = nroColumna(coordenada.charAt(0));
226         fila = Movimiento.movFila(fila, movimiento);
227         columna = Movimiento.movColumna(columna, movimiento);
228         System.out.println(fila + " Movimiento aplicado " + columna);
229         if(validezMovimiento(fila, columna)){
230             break;
231         } else {
232             System.out.println("Movimiento invalido");
233             fila = Movimiento.restFila(fila, movimiento);
234             columna = Movimiento.restColumna(columna, movimiento);
235         }
236     }
237     return movimiento;
238 }
239 public static boolean validezMovimiento(int fila, int columna){
240     return fila <= 9 && fila >= 0 && columna <= 9 && columna >= 0;
241 }
242 public static void datosEjercito(int n, ArrayList<Soldado> ejercito, char fig){
243     int nroSoldados = (int)(Math.random() * 10 + 1);
244     System.out.println("El ejercito " + n + " tiene un total de " + nroSoldados + " soldados");
245     for(int i = 0; i < nroSoldados; i++){
246         String nombre = "Soldado " + (i + 1) + "X" + n;
247         ejercito.add(new Soldado(nombre, fig));
248     }
249 }
250 public static void impTablero(char[][] tablero){

```

```

251 System.out.println("El tablero es :");
252 for(int x = 0; x < tablero.length; x++){
253     for(int y = 0; y < tablero[x].length; y++){
254         System.out.print(tablero[x][y]);
255     }
256     System.out.println();
257 }
258 }
259 public static void actTablero(ArrayList<Soldado> ejercito1, ArrayList<Soldado> ejercito2, char[][] tablero){
260     int fila;
261     char columna;
262     for(int i = 0; i < ejercito1.size(); i++){
263         fila = ejercito1.get(i).getFila();
264         columna = ejercito1.get(i).getColumna();
265         while(tablero[fila - 1][nroColumna(columna)] != '-'){
266             fila = (int)(Math.random() * 10 + 1);
267             columna = Soldado.numCol();
268         }
269         ejercito1.get(i).setFila(fila);
270         ejercito1.get(i).setCol(columna);
271         tablero[fila - 1][nroColumna(columna)] = ejercito1.get(i).getFigura();
272     }
273     for(int j = 0; j < ejercito2.size(); j++){
274         fila = ejercito2.get(j).getFila();
275         columna = ejercito2.get(j).getColumna();
276         while(tablero[fila - 1][nroColumna(columna)] != '-'){
277             fila = (int)(Math.random() * 10 + 1);
278             columna = Soldado.numCol();
279         }
280         ejercito2.get(j).setFila(fila);
281         ejercito2.get(j).setCol(columna);
282         tablero[fila - 1][nroColumna(columna)] = ejercito2.get(j).getFigura();
283     }
284 }
285 public static int nroColumna(char n){
286     switch(n){
287         case 'A': return 0;
288         case 'B': return 1;
289         case 'C': return 2;
290         case 'D': return 3;
291         case 'E': return 4;
292         case 'F': return 5;
293         case 'G': return 6;
294         case 'H': return 7;
295         case 'I': return 8;
296         case 'J': return 9;
297         default: return 0;
298     }
299 }
300 public static char conversionBusqueda(int m){
301     switch(m){
302         case 0: return 'A';
303         case 1: return 'B';
304         case 2: return 'C';
305         case 3: return 'D';
306         case 4: return 'E';
307         case 5: return 'F';
308         case 6: return 'G';
309         case 7: return 'H';
310         case 8: return 'I';
311         case 9: return 'J';
312         default: return 0;
313     }
314 }
315 public static int buscadorPosicion(ArrayList<Soldado> ejercito, String posicion){
316     for(int i = 0; i < ejercito.size(); i++){
317         if(posicion.charAt(0) == ejercito.get(i).getColumna() && Integer.parseInt(posicion.substring(1, posicion.length())) == i)
318             return i;
319     }



```

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

```

319     }
320 }
321 return 0;
322 }
323 //Metodos antiguos
324 public static char[][] tableroCopia(char[][] tablero){
325     char[][] tableroNuevo = new char[tablero.length][tablero[0].length];
326     for(int i = 0; i < tablero.length; i++){
327         for(int j = 0; j < tablero[0].length; j++){
328             tableroNuevo[i][j] = tablero[i][j];
329         }
330     }
331     return tableroNuevo;
332 }
333 public static void mayorVida(ArrayList<Soldado> ejercito, int n){
334     int mayor = ejercito.get(0).getVida();
335     for(int i = 1; i < ejercito.size(); i++){
336         if(ejercito.get(i).getVida() > mayor)
337             mayor = ejercito.get(i).getVida();
338     }
339     System.out.println("Los soldados con mayor vida del ejercito " + n + " son:");
340     for(int j = 0; j < ejercito.size(); j++){
341         if(ejercito.get(j).getVida() == mayor){
342             System.out.println(ejercito.get(j));
343         }
344     }
345 }
346 public static void promVida(ArrayList<Soldado> ejercito, int m){
347     double n = 0;
348     for(int i = 0; i < ejercito.size(); i++)
349         n += ejercito.get(i).getVida();
350     n /= ejercito.size();
351     System.out.println("El promedio de vida del ejercito " + m + " es " + n);
352 }
353 public static void copiaDatos(ArrayList<Soldado> ejercito, ArrayList<Soldado> copia){
354     for(int i = 0; i < ejercito.size(); i++){
355         Soldado pos = ejercito.get(i);
356         Soldado c = new Soldado(pos.getNombre(), pos.getAtaque(), pos.getDefensa(), pos.getVida(), pos.getNivVidAct(), pos.getVidAct());
357         copia.add(c);
358     }
359 }
360 public static void ordenarSeleccion(ArrayList<Soldado> ejercito){
361     for(int i = 0; i < ejercito.size() - 1; i++){

```

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

```

362         for(int j = i + 1; j < ejercito.size(); j++){
363             Soldado may = ejercito.get(i);
364             Soldado men = ejercito.get(j);
365             if(men.getVida() > may.getVida()){
366                 Soldado mayor = new Soldado(men.getNombre(), men.getAtaque(), men.getDefensa(), men.getVida(), men.getNivVidAct(
367                 Soldado menor = new Soldado(may.getNombre(), may.getAtaque(), may.getDefensa(), may.getVida(), may.getNivVidAct(
368                 ejercito.set(i, mayor);
369                 ejercito.set(j, menor);
370             }
371         }
372     }
373     mostDatos(ejercito);
374 }
375 public static void ordenarBurbuja(ArrayList<Soldado> ejercito){
376     for(int i = 0; i < ejercito.size() - 1; i++){
377         for(int j = 0; j < ejercito.size() - 1; j++){
378             Soldado may = ejercito.get(i);
379             Soldado men = ejercito.get(i + 1);
380             if(men.getVida() > may.getVida()){
381                 Soldado mayor = new Soldado(men.getNombre(), men.getAtaque(), men.getDefensa(), men.getVida(), men.getNivVidAct(
382                 Soldado menor = new Soldado(may.getNombre(), may.getAtaque(), may.getDefensa(), may.getVida(), may.getNivVidAct(
383                 ejercito.set(j, mayor);
384                 ejercito.set(j + 1, menor);
385             }
386         }
387     }
388     mostDatos(ejercito);
389 }
390 public static void mostDatos(ArrayList<Soldado> ejercito){
391     for(Soldado n: ejercito)
392         System.out.println(n);
393 }
394 }

```

	<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: 15

```

<terminated> VideoJuego (1) [Java Application] C:\Users\UJKMjuandi\Downloads\eclipse-java-2023-03-R-win32-x86_64\eclipse\
El ejercito 1 tiene un total de 3 soldados
El ejercito 2 tiene un total de 2 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
El tablero es :
*-----
&-----
-----
-----
-----

```


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

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 Topicos 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.

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	2	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1	

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 18</p>

4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	1	
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	2	
TOTAL		20		12	

Tabla 2: Rúbrica para contenido del Informe y demostración