	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: 1

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 Ejército - Soldado</i>				
NÚMERO DE PRÁCTICA:	<i>15</i>	AÑO LECTIVO:	<i>2023</i>	NRO. SEMESTRE:	<i>2do Semestre</i>
FECHA DE PRESENTACIÓN	<i>27/12/2023</i>	HORA DE PRESENTACIÓN	<i>18/40/00</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:

```

1 import java.util.ArrayList;
2 public class Ejercito {
3     private String nombreReino;
4     private ArrayList<Soldado> misSoldados;
5
6     public Ejercito(String nombreReino) {
7         this.nombreReino = nombreReino;
8         misSoldados = new ArrayList<>();
9     }
10    public void crearEjercito() {
11        int numSoldados = (int) (Math.random() * 10) + 1;
12        for (int i = 0; i < numSoldados; i++) {
13            Soldado soldado = new Soldado();
14            soldado.setNombre("Soldado " + (i + 1));
15            soldado.setNivelAtaque((int) (Math.random() * 5) + 1);
16            soldado.setNivelDefensa((int) (Math.random() * 5) + 1);
17            soldado.setVidaActual((int) (Math.random() * 5) + 1);
18            soldado.setVelocidad((int) (Math.random() * 5) + 1);
19            soldado.setActitud("ofensiva");
20            soldado.setFila((int) (Math.random() * 10) + 1);
21            soldado.setColumna((int) (Math.random() * 10) + 1);
22            soldado.setNivelVida((int) (Math.random() * 5) + 1);
23            agregarSoldado(soldado);
24        }
25    }
26    public void agregarSoldado(Soldado soldado) {
27        if (misSoldados.size() < 10) {
28            misSoldados.add(soldado);
29        } else {
30            System.out.println("El ejército está completo. No se pueden agregar más soldados.");
31        }
32    }
33
34    public void eliminarSoldado(int indice) {
35        if (indice >= 0 && indice < misSoldados.size()) {
36            misSoldados.remove(indice);
37        } else {
38            System.out.println("Índice de soldado no válido.");
39        }
40    }
41
42    public void modificarSoldado(int indice, String nombre, int nivelAtaque, int nivelDefensa,
43                                int velocidad, int fila, int columna, int nivelVida) {
44        if (indice >= 0 && indice < misSoldados.size()) {
45            Soldado soldado = misSoldados.get(indice);
46            soldado.setNombre(nombre);
47            soldado.setNivelVida(nivelVida);
48            soldado.setNivelAtaque(nivelAtaque);
49            soldado.setNivelDefensa(nivelDefensa);
50            soldado.setVelocidad(velocidad);
51            soldado.setFila(fila);
52            soldado.setColumna(columna);
53        } else {
54            System.out.println("Índice de soldado no válido.");
55        }
56    }
57    public void crearSoldado(String nombre, int nivelVida, int nivelAtaque, int nivelDefensa,
58                            int velocidad, int fila, int columna) {
59        Soldado soldado = new Soldado();
60        soldado.setNombre(nombre);
61        soldado.setNivelVida(nivelVida);
62        soldado.setNivelAtaque(nivelAtaque);
63        soldado.setNivelDefensa(nivelDefensa);
64        soldado.setVelocidad(velocidad);
65        soldado.setFila(fila);
66        soldado.setColumna(columna);
67        agregarSoldado(soldado);
68    }
69    public Soldado getSoldadoConMayorAtaque() {
70        Soldado soldadoConMayorAtaque = null;
71        int maxAtaque = -1;
72
73        for (Soldado soldado : misSoldados) {
74            if (soldado.getNivelAtaque() > maxAtaque) {
75                maxAtaque = soldado.getNivelAtaque();
76                soldadoConMayorAtaque = soldado;
77            }
78        }
79
80        return soldadoConMayorAtaque;
81    }
82    public void verRankingPoder() {
83        System.out.println("Ranking de poder por nivel de vida:");
84        int n = misSoldados.size();
85        boolean ordenado;
86        for (int i = 0; i < n - 1; i++) {

```

```

87         ordenado = false;
88         for (int j = 0; j < n - 1 - i; j++) {
89             if (misSoldados.get(j).getNivelVida() < misSoldados.get(j + 1).getNivelVida()) {
90                 Soldado temp = misSoldados.get(j);
91                 misSoldados.set(j, misSoldados.get(j + 1));
92                 misSoldados.set(j + 1, temp);
93                 ordenado = true;
94             }
95         }
96         if (!ordenado) {
97             break;
98         }
99     }
100     for (int i = 0; i < misSoldados.size(); i++) {
101         System.out.println("Posición " + (i + 1) + ": " + misSoldados.get(i).getNombre() + " - Vida: " + misSoldados.get(i).getNivelVida());
102     }
103 }
104 public String toString() {
105     String result = "Ejército del Reino: " + nombreReino + "\n";
106     for (Soldado soldado : misSoldados) {
107         result += "Nombre: " + soldado.getNombre() + "\n";
108         result += "Nivel de Ataque: " + soldado.getNivelAtaque() + "\n";
109         result += "Nivel de Defensa: " + soldado.getNivelDefensa() + "\n";
110         result += "Vida Actual: " + soldado.getVidaActual() + "\n";
111         result += "Velocidad: " + soldado.getVelocidad() + "\n";
112         result += "Fila: " + soldado.getFila() + "\n";
113         result += "Columna: " + soldado.getColumna() + "\n";
114         result += "Nivel de Vida: " + soldado.getNivelVida() + "\n";
115         result += "\n";
116     }
117     return result;
118 }
119 public ArrayList<Soldado> getMisSoldados() {
120     return misSoldados;
121 }
122 }

1 public class Soldado {
2     private String nombre;
3     private int nivelAtaque;
4     private int nivelDefensa;
5     private int vidaActual;
6     private int velocidad;
7     private String actitud;
8     private boolean vive;
9     private int fila;
10    private int columna;
11    private int nivelVida;
12
13    public void setNombre(String n){
14        nombre = n;
15    }
16    public void setNivelAtaque(int ataque){
17        nivelAtaque = ataque;
18    }
19    public void setNivelDefensa(int defensa){
20        nivelDefensa = defensa;
21    }
22    public void setVidaActual(int vidaAc) {
23        vidaActual = vidaAc;
24    }
25    public void setFila(int f){
26        fila = f;
27    }
28    public void setColumna(int c){
29        columna = c;
30    }
31    public void setNivelVida(int vida){
32        nivelVida = vida;
33    }
34    public void setVelocidad(int veloci){
35        velocidad = veloci;
36    }
37    public void setActitud(String ac) {
38        actitud = ac;
39    }
40    public String getNombre(){
41        return nombre;
42    }
43    public int getFila(){

```

```
44     return fila;
45 }
46 public int getColumna(){
47     return columna;
48 }
49 public int getNivelVida(){
50     return nivelVida;
51 }
52 public int getNivelAtaque(){
53     return nivelAtaque;
54 }
55 public int getNivelDefensa(){
56     return nivelDefensa;
57 }
58 public int getVelocidad(){
59     return velocidad;
60 }
61 public int getVidaActual() {
62     return vidaActual;
63 }
64 public void atacar() {
65     avanzar();
66     actitud = "ofensiva";
67 }
68 public void defender() {
69     velocidad = 0;
70     actitud = "defensiva";
71 }
72 public void avanzar() {
73     velocidad++;
74 }
75 public void retroceder() {
76     if (velocidad > 0) {
77         velocidad = 0;
78         actitud = "defensiva";
79     } else {
80         velocidad--;
81     }
82 }
83 public void serAtacado(int puntosDaño) {
84     vidaActual -= puntosDaño;
85     if (vidaActual <= 0) {
86         morir();
87     }
88 }
89 public void huir() {
90     velocidad += 2;
91     actitud = "fuga";
92 }
93 public void morir() {
94     vive = false;
95 }
96 }
```

```

1 import java.util.*;
2 public class VideoJuego {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         boolean salir = false;
6         System.out.print("Ingrese el nombre del reino que desea escoger (Inglaterra, Francia, Castilla-Aragón, Sacro\r\n"
7             + "Imperio Romano-Germánico, Moros): ");
8         String nombreReino = sc.nextLine();
9         Ejercito ejercito = new Ejercito(nombreReino);
10        boolean ejercitoCreado = false;
11        boolean continuarCreando = false;
12        while (!salir) {
13            System.out.println("\nMenú de opciones:");
14            if (ejercitoCreado) {
15                System.out.println("4. Eliminar soldado");
16                System.out.println("5. Modificar soldado");
17                System.out.println("6. Imprimir datos de los soldados");
18                System.out.println("7. Soldado con mayor nivel de vida");
19                System.out.println("8. Imprimir soldados por nivel de vida descendente");
20                System.out.println("9. Salir");
21            } else {
22                System.out.println("1. Crear ejército aleatorio");
23                System.out.println("2. Crear ejército manualmente");
24                System.out.println("3. Salir");
25            }
26
27            System.out.print("Elija una opción: ");
28            int opcion = sc.nextInt();
29            sc.nextLine();
30
31            switch (opcion) {
32                case 1:
33                    ejercito.crearEjercito();
34                    ejercitoCreado = true;
35                    break;
36                case 2:
37                    System.out.println("creando su ejercito:");
38                    System.out.println("crear soldado:");
39                    ArrayList<Soldado> soldados = ejercito.getMisSoldados();
40                    continuarCreando = true;
41                    while (continuarCreando && soldados.size() < 10) {
42                        System.out.print("Nuevo nombre: ");
43                        String nuevoNombre = sc.next();
44                        System.out.print("Nuevo nivel de ataque: ");
45                        int nuevoNivelAtaque = sc.nextInt();
46                        System.out.print("Nuevo nivel de defensa: ");
47                        int nuevoNivelDefensa = sc.nextInt();
48                        System.out.print("Nueva velocidad: ");
49                        int nuevaVelocidad = sc.nextInt();
50                        sc.nextLine();
51                        System.out.print("Nueva fila: ");
52                        int nuevaFila = sc.nextInt();
53                        System.out.print("Nueva columna: ");
54                        int nuevaColumna = sc.nextInt();
55                        System.out.print("Nuevo nivel de vida: ");
56                        int nuevoNivelVida = sc.nextInt();
57                        ejercito.crearSoldado(nuevoNombre, nuevoNivelAtaque, nuevoNivelDefensa, nuevaVelocidad, nuevaFila, nuevaColumna, nuevoNivelVida);
58                        if (soldados.size() < 10) {
59                            System.out.print("¿Desea crear otro soldado? (Si/No): ");
60                            String respuesta = sc.next().toLowerCase();
61                            if (respuesta.equals("no")) {
62                                continuarCreando = false;
63                                ejercitoCreado = true;
64                            }
65                        }
66                    }
67                    break;
68                case 3:
69                    System.out.println("Saliendo del programa.");
70                    salir = true;
71                    break;
72                case 4:
73                    System.out.print("Seleccione el número del soldado que desea eliminar: ");
74                    int numeroSoldado = sc.nextInt();
75                    ejercito.eliminarSoldado(numeroSoldado - 1);
76                    System.out.println("Soldado eliminado correctamente.");
77                    break;
78                case 5:
79                    ArrayList<Soldado> soldados = ejercito.getMisSoldados();
80                    System.out.print("Seleccione el número del soldado que desea modificar: ");
81                    int solda = sc.nextInt();
82                    if (solda >= 1 && solda <= soldados.size()) {
83                        sc.nextLine();
84                        System.out.print("Nuevo nombre: ");
85                        String nuevoNombre = sc.nextLine();
86                        System.out.print("Nuevo nivel de ataque: ");

```


	<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: 6

```

87         int nuevoNivelAtaque = sc.nextInt();
88         System.out.print("Nuevo nivel de defensa: ");
89         int nuevoNivelDefensa = sc.nextInt();
90         System.out.print("Nueva velocidad: ");
91         int nuevaVelocidad = sc.nextInt();
92         sc.nextLine();
93         System.out.print("Nueva fila: ");
94         int nuevaFila = sc.nextInt();
95         System.out.print("Nueva columna: ");
96         int nuevaColumna = sc.nextInt();
97         System.out.print("Nuevo nivel de vida: ");
98         int nuevoNivelVida = sc.nextInt();
99
100         ejercito.modificarSoldado(solda - 1, nuevoNombre, nuevoNivelAtaque, nuevoNivelDefensa, nuevaVelocidad, nuevaFila, nuevaColumna, nuev
101         System.out.println("Soldado modificado correctamente.");
102     } else {
103         System.out.println("Número de soldado no válido.");
104     }
105     break;
106 case 6:
107     System.out.println(ejercito.toString());
108     break;
109 case 7:
110     Soldado soldadoConMayorAtaque = ejercito.getSoldadoConMayorAtaque();
111     System.out.println("Soldado con mayor nivel de ataque: " + soldadoConMayorAtaque.getNombre());
112     System.out.println("Nivel de ataque: " + soldadoConMayorAtaque.getNivelAtaque());
113     break;
114 case 8:
115     ejercito.verRankingPoder();
116
117     break;
118 case 9:
119     System.out.println("Saliendo del programa.");
120     salir = true;
121     break;
122 default:
123     System.out.println("Opción no válida. Intente nuevamente.");
124 }
125 }
126 }
127 }

```

II. PRUEBA

	<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: 7</p>

Ingrese el nombre del reino que desea escoger (Inglaterra, Francia, Cast Imperio Romano-Germánico, Moros): **Inglaterra**

Menú de opciones:

1. Crear ejército aleatorio
2. Crear ejército manualmente
3. Salir

Elija una opción: **1**

Menú de opciones:

4. Eliminar soldado
5. Modificar soldado
6. Imprimir datos de los soldados
7. Soldado con mayor nivel de vida
8. Imprimir soldados por nivel de vida descendente
9. Salir

Elija una opción: **6**

Ejército del Reino: Inglaterra

Nombre: Soldado 1

Nivel de Ataque: 5

Nivel de Defensa: 1

Vida Actual: 2

Velocidad: 3

Fila: 6

Columna: 2

Nivel de Vida: 2

Nombre: Soldado 2

Nivel de Ataque: 5

Nivel de Defensa: 1

Vida Actual: 2

Velocidad: 2

Fila: 10

Columna: 3

Nivel de Vida: 4

Nombre: Soldado 3


Nivel de Ataque: 5

Nivel de Defensa: 4

Vida Actual: 4

Velocidad: 5

Fila: 5

	<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: 8</p>

III. CUESTIONARIO:

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

CONCLUSIONES

Un componente esencial de estas mejoras es la introducción de una clase principal denominada "Jugador", que permite a los usuarios crear y gestionar un ejército a través de un menú interactivo. Esta clase incluye atributos como Nombre y una lista de Soldados, donde cada soldado, representado por la clase "Soldado", tiene atributos como NivelAtaque y Vida. La gestión de eventos y resultados se ha afinado utilizando atributos como vidaActual, permitiendo una evaluación más precisa de las fortalezas y debilidades de los jugadores.


La interfaz de usuario ha sido mejorada para ofrecer una presentación clara de la información, facilitando la toma de decisiones estratégicas. Además, se han creado métodos adicionales en las clases pertinentes, como SeleccionarSoldado(), PlanificarMovimiento(), GestionarTurno(), Atacar(), RecibirDanio(), y EstablecerNivelAtaque(), para potenciar la interactividad y estrategia en el juego.

METODOLOGÍA DE TRABAJO

- 1.- *Leer los enunciados cuidadosamente para saber que requiere el problema.*
- 2.- *Seguir el diagrama uml como base que se da en la practica*
- 3.- *Empezar a hacer la ejercito, la clase soldado y la clase VideoJuego*
- 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

	<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: 9

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	
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	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	1	
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		15	


	<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: 10</p>

Tabla 2: Rúbrica para contenido del Informe y demostración