


	<p style="text-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>	
<p style="text-align: center;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2022/03/01</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p><b>Página:</b> 1</p>

## INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
<b>ASIGNATURA:</b>	<i>Fundamentos de la Programación 02</i>				
<b>TÍTULO DE LA PRÁCTICA:</b>	<i>HashMap</i>				
<b>NÚMERO DE PRÁCTICA:</b>	<i>08</i>	<b>AÑO LECTIVO:</b>	<i>2024</i>	<b>NRO. SEMESTRE:</b>	<i>2</i>
<b>FECHA DE PRESENTACIÓN</b>	<i>29/11/2024</i>	<b>HORA DE PRESENTACIÓN</b>	<i>17/30/00 PM</i>		
<b>INTEGRANTE (s)</b> <i>German Arturo Chipana Jerónimo</i>				<b>NOTA (0-20)</b>	
<b>DOCENTE(s):</b> <i>Pinto Oppe Lino José</i>					

RESULTADOS Y PRUEBAS
<p><b>I. EJERCICIOS RESUELTOS:</b></p> <p><i>El estudiante coloca la evidencia de los ejercicios propuestos realizados en la sesión de laboratorio, en el tiempo o duración indicado por el docente.</i></p> <p><i>El docente debe colocar la retroalimentación por cada ejercicio que el estudiante/grupo ha presentado</i></p> <p><b>EJERCICIO 01:</b></p> <ol style="list-style-type: none"> <li>1. Cree un Proyecto llamado Laboratorio8</li> <li>2. Usted deberá crear las dos clases Soldado.java y VideoJuego5.java. Puede reutilizar lo desarrollado en Laboratorios anteriores.</li> <li>3. Del Soldado nos importa el nombre, puntos de vida, fila y columna (posición en el tablero).</li> <li>4. El juego se desarrollará en el mismo tablero de los laboratorios anteriores. Para crear el tablero utilice la estructura de datos más adecuada.</li> <li>5. Tendrá 2 Ejércitos (usar HashMaps). Inicializar el tablero con n soldados aleatorios entre 1 y 10 para cada Ejército. Cada soldado tendrá un nombre autogenerado: Soldado0X1, Soldado1X1, etc., un valor de puntos de vida autogenerado aleatoriamente [1..5], la fila y columna también autogenerados aleatoriamente (no puede haber 2 soldados en el mismo cuadrado). Se debe mostrar el tablero con todos los soldados creados (distinguir los de un ejército de los del otro ejército). Además de los datos del Soldado con mayor vida de cada ejército, el promedio de puntos de vida de todos los soldados creados por ejército, los datos de todos los soldados por ejército en el orden que fueron creados y un ranking de poder de todos los soldados creados por ejército (del que tiene más nivel de vida al que tiene menos) usando 2 diferentes algoritmos de ordenamiento (indicar conclusiones respecto a este ordenamiento de HashMaps). Finalmente, que muestre qué ejército</li> </ol>

	<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> 2

ganará la batalla (indicar la métrica usada para decidir al ganador de la batalla). Hacerlo como programa iterativo.

### MAIN:

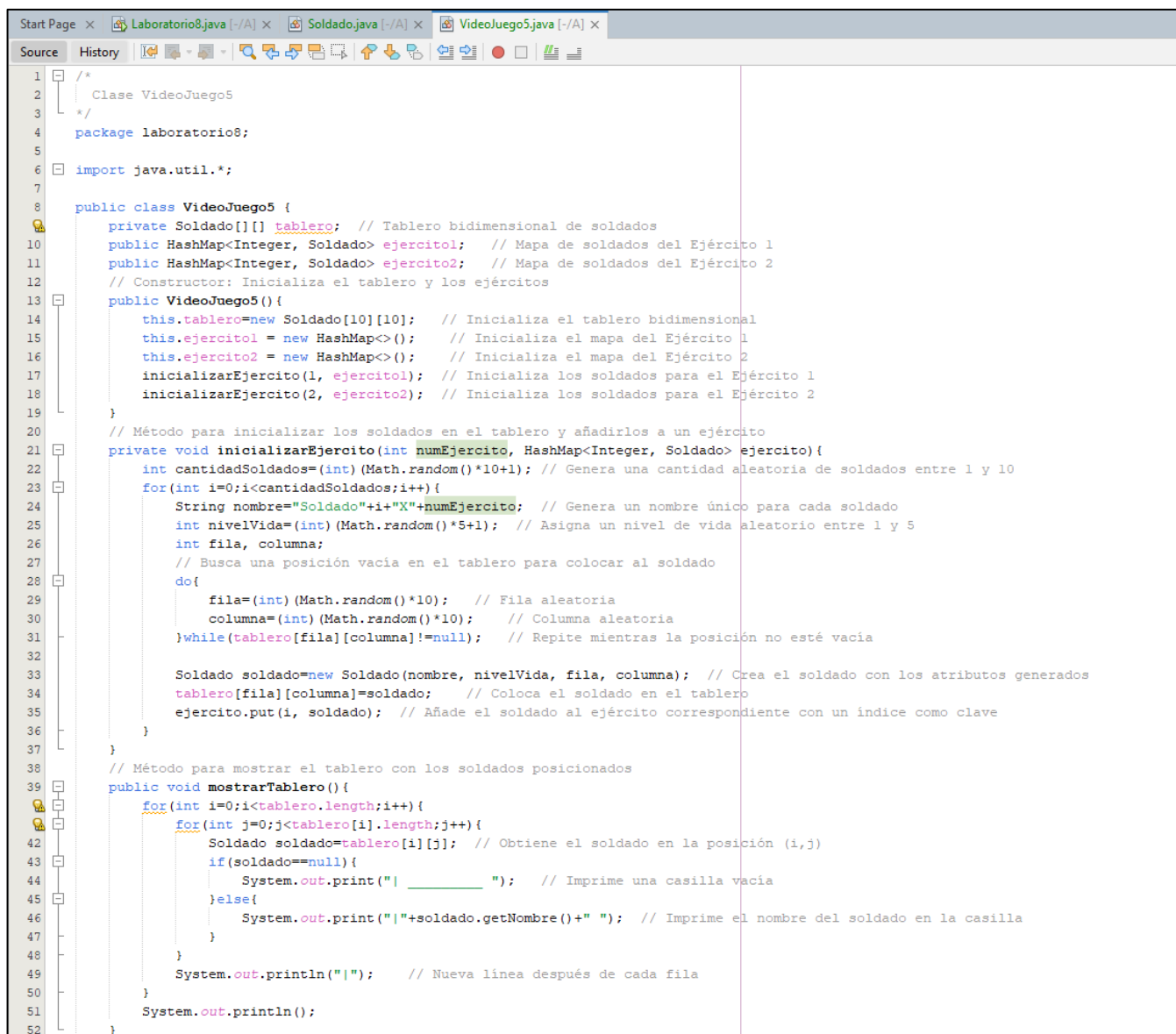
```

Start Page x Laboratorio8.java [-/A] x Soldado.java [-/A] x VideoJuego5.java [-/A] x
Source History
1  /*
2     Autor : Chipana Jeronimo German Arturo
3     Proposito : Laboratorio 8
4  */
5  package laboratorio8;
6
7  import java.util.*;
8
9  public class Laboratorio8 {
10
11     public static void main(String[] args) {
12         Scanner scan=new Scanner(System.in);
13         String respuesta="s";
14         while("s".equals(respuesta)){
15             VideoJuego5 juego=new VideoJuego5(); // Crea una instancia del juego con un tablero de 10x10
16             juego.mostrarTablero(); // Muestra el tablero actual con los soldados colocados
17             juego.mostrarSoldadosMayorVida(1, juego.ejercito1); // Muestra el soldado con mayor nivel de vida del Ejército 1
18             juego.mostrarSoldadosMayorVida(2, juego.ejercito2); // Muestra el soldado con mayor nivel de vida del Ejército 2
19             juego.mostrarPromedioNivelVida(1, juego.ejercito1); // Calcula y muestra el promedio de nivel de vida del Ejército 1
20             juego.mostrarPromedioNivelVida(2, juego.ejercito2); // Calcula y muestra el promedio de nivel de vida del Ejército 2
21             juego.mostrarSoldadosOrdenCreacion(1, juego.ejercito1); // Muestra los soldados del Ejército 1 en el orden en que fueron creados
22             juego.mostrarSoldadosOrdenCreacion(2, juego.ejercito2); // Muestra los soldados del Ejército 2 en el orden en que fueron creados
23             juego.mostrarRankingSoldados(1, juego.ejercito1); // Ordena y muestra el ranking de soldados del Ejército 1
24             juego.mostrarRankingSoldados(2, juego.ejercito2); // Ordena y muestra el ranking de soldados del Ejército 2
25             juego.mostrarEjercitoGanador(juego.ejercito1, juego.ejercito2); // Determina y muestra el ejército ganador
26             System.out.println("\nDesea generar una nueva batalla?(s/n)");
27             respuesta=scan.next();
28         }
29         System.out.println("\nGracias por jugar!");
30     }
31 }

```



**CLASE SOLDADO:**

```
Start Page x Laboratorio8.java [-/A] x Soldado.java [-/A] x VideoJuego5.java [-/A] x
Source History
1  /*
2  |  Clase Soldado
3  |  */
4  package laboratorio8;
5
6  public class Soldado {
7  |  private String nombre;
8  |  private int nivelVida;
9  |  private int fila;
10 |  private int columna;
11 |  //Constructor
12 |  public Soldado(String nombre, int nivelVida, int fila, int columna){
13 |  |  this.nombre = nombre;
14 |  |  this.nivelVida = nivelVida;
15 |  |  this.fila = fila;
16 |  |  this.columna = columna;
17 |  }
18 |  // Metodos accesorios
19 |  public String getNombre(){
20 |  |  return nombre;
21 |  }
22 |  public int getNivelVida(){
23 |  |  return nivelVida;
24 |  }
25 |  public int getFila(){
26 |  |  return fila;
27 |  }
28 |  public int getColumna(){
29 |  |  return columna;
30 |  }
31 |  // Metodo toString
32 |  public String toString(){
33 |  |  return "Soldado: "+nombre+", Vida: "+nivelVida+", Posicion: ("+(fila+1)+", "+(columna+1)+")";
34 |  }
35 }
```



```
// Método para mostrar el soldado con mayor nivel de vida en un ejército
53 public void mostrarSoldadosMayorVida(int numEjercito, HashMap<Integer, Soldado> ejercito){
54     Soldado mayorVida=null;
55     for(Soldado soldado : ejercito.values()){
56         if(mayorVida==null || soldado.getNivelVida().getNivelVida()>mayorVida.getNivelVida()){
57             mayorVida=soldado; // Actualiza si encuentra un soldado con mayor nivel de vida
58         }
59     }
60     System.out.println("Soldado con mayor nivel de vida del ejercito "+numEjercito+" : "+mayorVida);
61     System.out.println();
62 }
63
64 // Método para calcular y mostrar el promedio del nivel de vida de un ejército
65 public void mostrarPromedioNivelVida(int numEjercito, HashMap<Integer, Soldado> ejercito){
66     int sumaVida=0;
67     for(Soldado soldado : ejercito.values()){
68         sumaVida+=soldado.getNivelVida(); // Suma el nivel de vida de cada soldado
69     }
70     double promedio=(double)sumaVida/ejercito.size(); // Calcula el promedio
71     System.out.println("Promedio del nivel de vida del ejercito "+numEjercito+" : "+promedio);
72     System.out.println();
73 }
74
75 // Método para mostrar los soldados en el orden en que fueron creados
76 public void mostrarSoldadosOrdenCreacion(int numEjercito, HashMap<Integer, Soldado> ejercito){
77     System.out.println("Soldados del ejercito "+numEjercito+" en el orden de creacion:");
78     for(Soldado soldado : ejercito.values()){
79         System.out.println(soldado); // Imprime cada soldado en el orden de creación
80     }
81     System.out.println();
82 }
83
84 // Método para mostrar el ranking de soldados basado en su nivel de vida, usando burbuja y selección
85 public void mostrarRankingSoldados(int numEjercito, HashMap<Integer, Soldado> ejercito){
86     // Ordenamiento por burbuja
87     ArrayList<Soldado> ejercitoBurbuja=new ArrayList<>(ejercito.values()); // Convierte el HashMap a lista
88     burbujaOrdenar(ejercitoBurbuja); // Ordena usando el algoritmo de burbuja
89     System.out.println("Ranking de soldados ejercito "+numEjercito+" (Burbuja): ");
90     for(Soldado soldado : ejercitoBurbuja){
91         System.out.println(soldado); // Imprime los soldados ordenados
92     }
93     // Ordenamiento por selección
94     ArrayList<Soldado> ejercitoSeleccion=new ArrayList<>(ejercito.values()); // Convierte el HashMap a lista
95     seleccionOrdenar(ejercitoSeleccion); // Ordena usando el algoritmo de selección
96     System.out.println("Ranking de soldados ejercito "+numEjercito+" (Selección): ");
97     for(Soldado soldado : ejercitoSeleccion){
98         System.out.println(soldado); // Imprime los soldados ordenados
99     }
100     System.out.println();
101 }
```

```
100 // Método de ordenamiento por burbuja para ordenar soldados por nivel de vida (descendente)
101 private void burbujaOrdenar(ArrayList<Soldado> soldados){
102     for(int i=0;i<soldados.size()-1;i++){
103         for(int j=0;j<soldados.size()-1-i;j++){
104             if(soldados.get(j).getNivelVida()<soldados.get(j+1).getNivelVida()){
105                 // Intercambia los soldados si el nivel de vida del siguiente es mayor
106                 Soldado temp=soldados.get(j);
107                 soldados.set(j, soldados.get(j+1));
108                 soldados.set(j+1, temp);
109             }
110         }
111     }
112 }
113 // Método de ordenamiento por selección para ordenar soldados por nivel de vida (descendente)
114 private void seleccionOrdenar(ArrayList<Soldado> soldados){
115     for(int i=0;i<soldados.size()-1;i++){
116         int maxIndex=i; // Asume que el soldado en la posición i tiene el mayor nivel de vida
117         for(int j=i+1;j<soldados.size();j++){
118             if(soldados.get(j).getNivelVida()>soldados.get(maxIndex).getNivelVida()){
119                 maxIndex=j; // Actualiza la posición del soldado con mayor nivel de vida
120             }
121         }
122         // Intercambia el soldado en i con el de mayor nivel de vida encontrado
123         Soldado temp=soldados.get(i);
124         soldados.set(i, soldados.get(maxIndex));
125         soldados.set(maxIndex, temp);
126     }
127 }
128 // Método que determina y muestra el ejército ganador basado en el número de soldados
129 public void mostrarEjercitoGanador(HashMap<Integer, Soldado> ejercito1, HashMap<Integer, Soldado> ejercito2){
130     if(ejercito1.size()>ejercito2.size()){
131         if(ejercito1.size()==ejercito2.size()){
132             System.out.println("Hubo un EMPATE de ejércitos, ambos con "+ejercito1.size()+" soldados...");
133         }
134         else
135             System.out.println("Gano el EJERCITO 1 con "+ejercito1.size()+" soldados!");
136     }
137     else
138         System.out.println("Gano el EJERCITO 2 con "+ejercito2.size()+" soldados!");
139 }
```

	<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;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2022/03/01</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p><b>Página:</b> 7</p>

## II. PRUEBAS

*¿Con que valores comprobaste que tu práctica estuviera correcta?*

- **Tamaño del tablero:** Comprobé con un tablero bidimensional de 10x10 filas y columnas.
- **Cantidad de soldados:** La cantidad de soldados por ejército varía de 1 a 10 de manera aleatoria.

*¿Qué resultado esperabas obtener para cada valor de entrada?*

- **Tablero:** Esperaba ver un tablero de 10x10 con algunas casillas llenas de soldados (representados por nombres como "Soldado0X1") y otras vacías (con null).
- **Mayor nivel de vida:** Esperaba que se identificara correctamente el soldado con el mayor nivel de vida (valor entre 1 y 5) para cada ejército.
- **Promedio de nivel de vida:** Esperaba obtener un número decimal que reflejara el promedio del nivel de vida de los soldados de cada ejército.
- **Orden de creación:** Esperaba ver los soldados de cada ejército impresos en el orden en que fueron generados, con su nombre y nivel de vida.
- **Ranking de soldados:** Esperaba ver los soldados ordenados correctamente en dos rankings (burbuja y selección), ambos descendiendo por el nivel de vida.
- **Ejército ganador:** Esperaba que el ejército con más soldados fuera declarado ganador, o que se indicara un empate si ambos ejércitos tenían el mismo número de soldados.
- **Iterativo:** Esperaba que el programa pregunte si se desea generar otra batalla o no.

*¿Qué valor o comportamiento obtuviste para cada valor de entrada?*

- **Tablero:** Obtuve un tablero de 10x10, donde varias posiciones estaban ocupadas por soldados y otras vacías, como se esperaba.
- **Mayor nivel de vida:** El código identificó correctamente al soldado con el nivel más alto de vida para cada ejército.
- **Promedio de nivel de vida:** El promedio calculado para cada ejército fue correcto y reflejó los niveles de vida de los soldados creados aleatoriamente.
- **Orden de creación:** Los soldados fueron mostrados en el orden en que fueron creados, con su nombre, nivel de vida y posición en el tablero.
- **Ranking de soldados:** Los rankings obtenidos por los algoritmos de burbuja y selección coincidieron y ordenaron a los soldados de mayor a menor nivel de vida, como se esperaba.
- **Ejército ganador:** El ejército con más soldados fue declarado ganador, o se mostró un mensaje de empate si tenían la misma cantidad de soldados.
- **Iterativo:** Se mostró un mensaje para saber si se deseaba generar otra batalla o no.

## LABORATORIO8

### 1. Se mostró el tablero con los soldados de cada Ejército en su respectiva posición.

Output - Laboratorio8 (run) x

run:

								Soldado3X2			
				Soldado3X1							
		Soldado0X2								Soldado4X2	
			Soldado5X2	Soldado2X2							
		Soldado6X2								Soldado1X1	
			Soldado2X1					Soldado0X1			
		Soldado8X2									
			Soldado7X2				Soldado1X2				

### 2. Se mostró el soldado con mayor nivel de vida el Ejército 1.

Soldado con mayor nivel de vida del ejercito 1 : Soldado: Soldado0X1, Vida: 4, Posicion: (7, 8)

### 3. Se mostró el soldado con mayor nivel de vida el Ejército 2.

Soldado con mayor nivel de vida del ejercito 2 : Soldado: Soldado4X2, Vida: 5, Posicion: (4, 9)

### 4. Se mostró el promedio del nivel de vida de todos los soldados del Ejército 1.

Promedio del nivel de vida del ejercito 1 : 3.5



**5. Se mostró el promedio del nivel de vida de todos los soldados del Ejército 2.**

```
Promedio del nivel de vida del ejercito 2 : 2.6666666666666665
```

**6. Se mostró a todos los soldados creados del Ejército 1 por orden de creación.**

```
Soldados del ejercito 1 en el orden de creacion:
Soldado: Soldado0X1, Vida: 4, Posicion: (7, 8)
Soldado: Soldado1X1, Vida: 4, Posicion: (6, 9)
Soldado: Soldado2X1, Vida: 4, Posicion: (7, 3)
Soldado: Soldado3X1, Vida: 2, Posicion: (2, 4)
```

**7. Se mostró a todos los soldados creados del Ejército 2 por orden de creación.**

```
Soldados del ejercito 2 en el orden de creacion:
Soldado: Soldado0X2, Vida: 4, Posicion: (4, 2)
Soldado: Soldado1X2, Vida: 3, Posicion: (9, 7)
Soldado: Soldado2X2, Vida: 1, Posicion: (5, 4)
Soldado: Soldado3X2, Vida: 3, Posicion: (1, 8)
Soldado: Soldado4X2, Vida: 5, Posicion: (4, 9)
Soldado: Soldado5X2, Vida: 2, Posicion: (5, 3)
Soldado: Soldado6X2, Vida: 2, Posicion: (6, 2)
Soldado: Soldado7X2, Vida: 1, Posicion: (9, 3)
Soldado: Soldado8X2, Vida: 3, Posicion: (8, 2)
```

**8. Se ordenaron los soldados del Ejército 1 por nivel de vida usando el ORDENAMIENTO BURBUJA y ORDENAMIENTO SELECCIÓN.**

```

Ranking de soldados ejercito 1 (Burbuja):
Soldado: Soldado0X1, Vida: 4, Posicion: (7, 8)
Soldado: Soldado1X1, Vida: 4, Posicion: (6, 9)
Soldado: Soldado2X1, Vida: 4, Posicion: (7, 3)
Soldado: Soldado3X1, Vida: 2, Posicion: (2, 4)
Ranking de soldados ejercito 1 (Selección):
Soldado: Soldado0X1, Vida: 4, Posicion: (7, 8)
Soldado: Soldado1X1, Vida: 4, Posicion: (6, 9)
Soldado: Soldado2X1, Vida: 4, Posicion: (7, 3)
Soldado: Soldado3X1, Vida: 2, Posicion: (2, 4)



```

**9. Se ordenaron los soldados del Ejército 2 por nivel de vida usando el ORDENAMIENTO BURBUJA y ORDENAMIENTO SELECCIÓN.**

```

Ranking de soldados ejercito 2 (Burbuja):
Soldado: Soldado4X2, Vida: 5, Posicion: (4, 9)
Soldado: Soldado0X2, Vida: 4, Posicion: (4, 2)
Soldado: Soldado1X2, Vida: 3, Posicion: (9, 7)
Soldado: Soldado3X2, Vida: 3, Posicion: (1, 8)
Soldado: Soldado8X2, Vida: 3, Posicion: (8, 2)
Soldado: Soldado5X2, Vida: 2, Posicion: (5, 3)
Soldado: Soldado6X2, Vida: 2, Posicion: (6, 2)
Soldado: Soldado2X2, Vida: 1, Posicion: (5, 4)
Soldado: Soldado7X2, Vida: 1, Posicion: (9, 3)
Ranking de soldados ejercito 2 (Selección):
Soldado: Soldado4X2, Vida: 5, Posicion: (4, 9)
Soldado: Soldado0X2, Vida: 4, Posicion: (4, 2)
Soldado: Soldado3X2, Vida: 3, Posicion: (1, 8)
Soldado: Soldado1X2, Vida: 3, Posicion: (9, 7)
Soldado: Soldado8X2, Vida: 3, Posicion: (8, 2)
Soldado: Soldado5X2, Vida: 2, Posicion: (5, 3)
Soldado: Soldado6X2, Vida: 2, Posicion: (6, 2)
Soldado: Soldado7X2, Vida: 1, Posicion: (9, 3)
Soldado: Soldado2X2, Vida: 1, Posicion: (5, 4)

```

	<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: 11</p>

**10. Se mostró que Ejército ganó la batalla y con cuantos soldados.**

```
Gano el EJERCITO 2 con 9 soldados!
```

**11. Se mostró un mensaje para saber si el usuario desea generar otra batalla.**

```
Desea generar una nueva batalla?(s/n)
```

**12. Se mostró un mensaje de despedida si el usuario ingresaba (n).**

```
Desea generar una nueva batalla?(s/n)
n

Gracias por jugar!
BUILD SUCCESSFUL (total time: 2 minutes 15 seconds)
|
```

## 11. Ejecución completa.

```

Output - Laboratorio8 (run) x
run:
| _____ | _____ | _____ | _____ | _____ | _____ | _____ | Soldado3X2 | _____ | _____ |
| _____ | _____ | _____ | Soldado3X1 | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | Soldado0X2 | _____ | _____ | _____ | _____ | _____ | _____ | Soldado4X2 | _____ |
| _____ | _____ | Soldado5X2 | Soldado2X2 | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | Soldado6X2 | _____ | _____ | _____ | _____ | _____ | _____ | Soldado1X1 | _____ |
| _____ | _____ | Soldado2X1 | _____ | _____ | _____ | _____ | Soldado0X1 | _____ | _____ |
| _____ | Soldado8X2 | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | _____ | Soldado7X2 | _____ | _____ | _____ | Soldado1X2 | _____ | _____ | _____ |
| _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |

Soldado con mayor nivel de vida del ejercito 1 : Soldado: Soldado0X1, Vida: 4, Posicion: (7, 8)

Soldado con mayor nivel de vida del ejercito 2 : Soldado: Soldado4X2, Vida: 5, Posicion: (4, 9)

Promedio del nivel de vida del ejercito 1 : 3.5



Promedio del nivel de vida del ejercito 2 : 2.6666666666666665

Soldados del ejercito 1 en el orden de creacion:
Soldado: Soldado0X1, Vida: 4, Posicion: (7, 8)
Soldado: Soldado1X1, Vida: 4, Posicion: (6, 9)
Soldado: Soldado2X1, Vida: 4, Posicion: (7, 3)
Soldado: Soldado3X1, Vida: 2, Posicion: (2, 4)

Soldados del ejercito 2 en el orden de creacion:
Soldado: Soldado0X2, Vida: 4, Posicion: (4, 2)
Soldado: Soldado1X2, Vida: 3, Posicion: (9, 7)
Soldado: Soldado2X2, Vida: 1, Posicion: (5, 4)
Soldado: Soldado3X2, Vida: 3, Posicion: (1, 8)
Soldado: Soldado4X2, Vida: 5, Posicion: (4, 9)
Soldado: Soldado5X2, Vida: 2, Posicion: (5, 3)
Soldado: Soldado6X2, Vida: 2, Posicion: (6, 2)
Soldado: Soldado7X2, Vida: 1, Posicion: (9, 3)
Soldado: Soldado8X2, Vida: 3, Posicion: (8, 2)

Ranking de soldados ejercito 1 (Burbuja):
Soldado: Soldado0X1, Vida: 4, Posicion: (7, 8)
Soldado: Soldado1X1, Vida: 4, Posicion: (6, 9)
Soldado: Soldado2X1, Vida: 4, Posicion: (7, 3)
Soldado: Soldado3X1, Vida: 2, Posicion: (2, 4)
Ranking de soldados ejercito 1 (Seleccion):
Soldado: Soldado0X1, Vida: 4, Posicion: (7, 8)
Soldado: Soldado1X1, Vida: 4, Posicion: (6, 9)
Soldado: Soldado2X1, Vida: 4, Posicion: (7, 3)
Soldado: Soldado3X1, Vida: 2, Posicion: (2, 4)

```

	<p style="text-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>	
<p style="text-align: center;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2022/03/01</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p><b>Página:</b> 13</p>

```

Ranking de soldados ejercito 1 (Burbuja):
Soldado: Soldado0X1, Vida: 4, Posicion: (7, 8)
Soldado: Soldado1X1, Vida: 4, Posicion: (6, 9)
Soldado: Soldado2X1, Vida: 4, Posicion: (7, 3)
Soldado: Soldado3X1, Vida: 2, Posicion: (2, 4)
Ranking de soldados ejercito 1 (Seleccion):
Soldado: Soldado0X1, Vida: 4, Posicion: (7, 8)
Soldado: Soldado1X1, Vida: 4, Posicion: (6, 9)
Soldado: Soldado2X1, Vida: 4, Posicion: (7, 3)
Soldado: Soldado3X1, Vida: 2, Posicion: (2, 4)



Ranking de soldados ejercito 2 (Burbuja):
Soldado: Soldado4X2, Vida: 5, Posicion: (4, 9)
Soldado: Soldado0X2, Vida: 4, Posicion: (4, 2)
Soldado: Soldado1X2, Vida: 3, Posicion: (9, 7)
Soldado: Soldado3X2, Vida: 3, Posicion: (1, 8)
Soldado: Soldado8X2, Vida: 3, Posicion: (8, 2)
Soldado: Soldado5X2, Vida: 2, Posicion: (5, 3)
Soldado: Soldado6X2, Vida: 2, Posicion: (6, 2)
Soldado: Soldado2X2, Vida: 1, Posicion: (5, 4)
Soldado: Soldado7X2, Vida: 1, Posicion: (9, 3)
Ranking de soldados ejercito 2 (Seleccion):
Soldado: Soldado4X2, Vida: 5, Posicion: (4, 9)
Soldado: Soldado0X2, Vida: 4, Posicion: (4, 2)
Soldado: Soldado3X2, Vida: 3, Posicion: (1, 8)
Soldado: Soldado1X2, Vida: 3, Posicion: (9, 7)
Soldado: Soldado8X2, Vida: 3, Posicion: (8, 2)
Soldado: Soldado5X2, Vida: 2, Posicion: (5, 3)
Soldado: Soldado6X2, Vida: 2, Posicion: (6, 2)
Soldado: Soldado7X2, Vida: 1, Posicion: (9, 3)
Soldado: Soldado2X2, Vida: 1, Posicion: (5, 4)

Gano el EJERCITO 2 con 9 soldados!

Desea generar una nueva batalla?(s/n)
n

Gracias por jugar!
BUILD SUCCESSFUL (total time: 2 minutes 15 seconds)

```

	<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;"><b>Formato:</b> 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>

## MIS COMMITS:

### COMMIT 1 (REUTILIZANDO LABORATORIO 07):

```

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git add LABORATORIO_08

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git commit -m "Reutilizando Laboratorio 07"
[main 38f7ee7] Reutilizando Laboratorio 07
 3 files changed, 205 insertions(+)
 create mode 100644 LABORATORIO_08/Laboratorio8.java
 create mode 100644 LABORATORIO_08/Soldado.java
 create mode 100644 LABORATORIO_08/VideoJuego5.java

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 3.15 KiB | 1.57 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ChipanaGerman/LABORATORIOS-FP2.git
   fe9e6d7..38f7ee7  main -> main

```

Se agregaron los archivos Laboratorio7.java, Soldado.java y VideoJuego4.java al repositorio de GitHub adaptados a las indicaciones del Laboratorio8.

### COMMIT 2 (ATRIBUTOS Y CONSTRUCTOR):



```

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git add LABORATORIO_08

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git commit -m "Modificando atributos y constructor para HashMap"
[main 4560b6f] Modificando atributos y constructor para HashMap
 1 file changed, 5 insertions(+), 5 deletions(-)

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 496 bytes | 496.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/ChipanaGerman/LABORATORIOS-FP2.git
   38f7ee7..4560b6f  main -> main

```

	<p style="text-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>	
<p style="text-align: center;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2022/03/01</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p><b>Página:</b> 15</p>

Modificando atributos y constructor para HashMap.

#### COMMIT 3 (INICIALIZACION DE EJERCITOS):

```
user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git add LABORATORIO_08

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git commit -m "Modificando la inicializacion de ejercitos"
[main fe4149b] Modificando la inicializacion de ejercitos
1 file changed, 1 insertion(+), 1 deletion(-)

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 408 bytes | 408.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/ChipanaGerman/LABORATORIOS_FP2.git
4560b6f..fe4149b main -> main
```

Modificando la inicialización de ejércitos.



#### COMMIT 4 (METODOS SOLDADO MAYOR VIDA Y PROMEDIO DE VIDA):

```
user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git add LABORATORIO_08

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git commit -m "Modificando los metodos de soldado con mayor vida y promedio de vida de cada ejercito"
[main dc37aae] Modificando los metodos de soldado con mayor vida y promedio de vida de cada ejercito
1 file changed, 6 insertions(+), 6 deletions(-)

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 472 bytes | 472.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/ChipanaGerman/LABORATORIOS_FP2.git
fe4149b..dc37aae main -> main
```

Modificando métodos de soldado con mayor vida y promedio de vida de cada ejército.

	<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;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 16</p>

#### COMMIT 5 (METODOS CREACION Y RANKING):

```

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git add LABORATORIO_08

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git commit -m "Modificando metodos de orden de creacion y ranking de soldados"
[main 711b3cb] Modificando metodos de orden de creacion y ranking de soldados
1 file changed, 5 insertions(+), 5 deletions(-)

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 488 bytes | 488.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/ChipanaGerman/LABORATORIOS-FP2.git
dc37aae..711b3cb main -> main

```

Modificando métodos de orden de creación y ranking de soldados de cada ejército.

#### COMMIT 6 (VERSION FINAL):

```

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git add LABORATORIO_08

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git commit -m "Modificando metodo de definir ganador y VERSION FINAL"
[main 8c537ca] Modificando metodo de definir ganador y VERSION FINAL
1 file changed, 1 insertion(+), 1 deletion(-)

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 398 bytes | 398.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/ChipanaGerman/LABORATORIOS-FP2.git
711b3cb..8c537ca main -> main



```

Modificando método de definir ganador y VERSION FINAL.



### III. RUBRICA:

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	1	
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	2	
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	2	
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	4	
TOTAL		20		18	

	<p style="text-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> 18

*Colocar la evidencia de las respuestas realizadas al cuestionario enunciado en la guía práctica de laboratorio.*

## CONCLUSIONES

*Colocar las conclusiones, apreciaciones reflexivas, opiniones finales a cerca de los resultados obtenidos de la sesión de laboratorio.*

### CONCLUSIÓN:

El uso de `HashMap` fue fundamental para gestionar eficientemente los ejércitos en el videojuego. Esta estructura permitió almacenar y acceder rápidamente a los soldados, utilizando claves únicas que facilitaron la identificación y manipulación de datos como niveles de vida y posiciones. Su flexibilidad para representar a los ejércitos como conjuntos dinámicos resultó clave para inicializar, ordenar y analizar la información de forma organizada. Además, al combinar `HashMap` con algoritmos de ordenamiento, se exploraron estrategias para convertir estos mapas en listas ordenadas, destacando las fortalezas de esta estructura en aplicaciones complejas como la simulación de batallas.

## METODOLOGÍA DE TRABAJO



*Colocar la metodología de trabajo que ha utilizado el estudiante o el grupo para resolver la práctica, es decir el procedimiento/secuencia de pasos en forma general.*

### 1. Análisis del Problema

- Se identificaron los requisitos del sistema, que incluyen la representación de soldados, un tablero, dos ejércitos, y las operaciones necesarias (inicialización, ordenamiento, cálculos y determinación del ganador).
- Se definieron las entidades principales: la clase `Soldado` y la clase `VideoJuego5`.

### 2. Diseño del Sistema

- **Clase `Soldado`:** Representa las propiedades y comportamiento de un soldado, como nombre, puntos de vida y posición en el tablero.
- **Clase `VideoJuego5`:** Incluye un tablero bidimensional para la disposición de soldados y dos estructuras de datos `HashMap` para representar los ejércitos. Los métodos implementados cubren todas las funcionalidades requeridas.

	<p style="text-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>	
<p style="text-align: center;"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2022/03/01</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p><b>Página:</b> 19</p>

### 3. Implementación

- Se creó el tablero como una matriz bidimensional para garantizar que no hubiera más de un soldado por casilla.
- Los `HashMap` se utilizaron para gestionar los ejércitos debido a su eficiencia en la búsqueda y almacenamiento de datos.
- Se generaron valores aleatorios para los atributos de los soldados (nombre, nivel de vida y posición en el tablero).
- Métodos adicionales como `mostrarTablero`, `mostrarRankingSoldados` y `mostrarEjercitoGanador` se diseñaron para cumplir con los requisitos funcionales y generar una salida detallada.

### 4. Ordenamiento

- Se implementaron dos algoritmos: Burbuja y Selección para ordenar los soldados por nivel de vida. Esto permitió analizar las diferencias en tiempo de ejecución y facilidad de implementación.

### 5. Pruebas

- Se probaron múltiples escenarios para garantizar que:
  - Los soldados fueran generados correctamente y no se duplicaran en el tablero.
  - Los cálculos del promedio y el ranking fueran precisos.
  - El programa identificara correctamente al ejército ganador.

## REFERENCIAS Y BIBLIOGRAFÍA

*Colocare las referencias utilizadas para el desarrollo de la práctica en formato IEEE*

M. Aedo López, Práctica de Laboratorio 8: HashMap, Universidad Nacional de San Agustín, 2023.  
<https://github.com/ChipanaGerman/LABORATORIOS-FP2>