


	<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: 1

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	<i>Fundamentos de la Programación 02</i>				
TÍTULO DE LA PRÁCTICA:	<i>Combinando Arreglos Estándar y ArrayList</i>				
NÚMERO DE PRÁCTICA:	<i>07</i>	AÑO LECTIVO:	<i>2024</i>	NRO. SEMESTRE:	<i>2</i>
FECHA DE PRESENTACIÓN	<i>15/11/2024</i>	HORA DE PRESENTACIÓN	<i>17/30/00 PM</i>		
INTEGRANTE (s) <i>German Arturo Chipana Jerónimo</i>				NOTA (0-20)	
DOCENTE(s): <i>Pinto Oppe Lino José</i>					

RESULTADOS Y PRUEBAS
<p>I. EJERCICIOS RESUELTOS:</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>EJERCICIO 01:</p> <ol style="list-style-type: none"> 1. Cree un Proyecto llamado Laboratorio7 2. Usted deberá crear las dos clases Soldado.java y VideoJuego4.java. Puede reutilizar lo desarrollado en Laboratorios anteriores. 3. Del Soldado nos importa el nombre, puntos de vida, fila y columna (posición en el tablero). 4. El juego se desarrollará en el mismo tablero de los laboratorios anteriores. Para el tablero utilizar la estructura de datos más adecuada. 5. Tendrá 2 Ejércitos (utilizar la estructura de datos más adecuada). 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 y sus puntos de vida (usar caracteres como _ y otros y 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. Finalmente, que

	<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: 2

muestre qué ejército ganará la batalla (indicar la métrica usada para decidir al ganador de la batalla).
 Hacer el programa iterativo.

MAIN:

```

1  /*
2  Autor : Chipana Jeronimo German Arturo
3  Proposito : Laboratorio 07
4  */
5  package laboratorio7;
6
7  import java.util.*;
8  public class Laboratorio7 {
9
10     public static void main(String[] args) {
11         Scanner scan=new Scanner(System.in);
12         String respuesta="s";
13         while("s".equals(respuesta)){
14             VideoJuego4 juego=new VideoJuego4(); // Crea una instancia del juego con un tablero de 10x10
15             juego.mostrarTablero(); // Muestra el tablero actual con los soldados colocados
16             juego.mostrarSoldadosMayorVida(1, juego.ejercito1); // Muestra el soldado con mayor nivel de vida del Ejército 1
17             juego.mostrarSoldadosMayorVida(2, juego.ejercito2); // Muestra el soldado con mayor nivel de vida del Ejército 2
18             juego.mostrarPromedioNivelVida(1, juego.ejercito1); // Calcula y muestra el promedio de nivel de vida del Ejército 1
19             juego.mostrarPromedioNivelVida(2, juego.ejercito2); // Calcula y muestra el promedio de nivel de vida del Ejército 2
20             juego.mostrarSoldadosOrdenCreacion(1, juego.ejercito1); // Muestra los soldados del Ejército 1 en el orden en que fueron creados
21             juego.mostrarSoldadosOrdenCreacion(2, juego.ejercito2); // Muestra los soldados del Ejército 2 en el orden en que fueron creados
22             juego.mostrarRankingSoldados(1, juego.ejercito1); // Ordena y muestra el ranking de soldados del Ejército 1
23             juego.mostrarRankingSoldados(2, juego.ejercito2); // Ordena y muestra el ranking de soldados del Ejército 2
24             juego.mostrarEjercitoGanador(juego.ejercito1, juego.ejercito2); // Determina y muestra el ejército ganador
25             System.out.println("\nDesea generar una nueva batalla?(s/n)");
26             respuesta=scan.next();
27         }
28         System.out.println("\nGracias por jugar!");
29     }
30 }
  
```

CLASE SOLDADO:



```
Start Page x Laboratorio7.java x Soldado.java x VideoJuego4.java x
Source History
1  /*
2  Clase Soldado
3  */
4  package laboratorio7;
5
6
7  public class Soldado {
8      private String nombre;
9      private int nivelVida;
10     private int fila;
11     private int columna;
12     //Constructor
13     public Soldado(String nombre, int nivelVida, int fila, int columna){
14         this.nombre = nombre;
15         this.nivelVida = nivelVida;
16         this.fila = fila;
17         this.columna = columna;
18     }
19     // Metodos accesorios
20     public String getNombre(){
21         return nombre;
22     }
23     public int getNivelVida(){
24         return nivelVida;
25     }
26     public int getFila(){
27         return fila;
28     }
29     public int getColumna(){
30         return columna;
31     }
32     // Metodo toString
33     public String toString(){
34         return "Soldado: "+nombre+", Vida: "+nivelVida+", Posicion: ("+(fila+1)+", "+(columna+1)+")";
35     }
36 }
```

CLASE VIDEOJUEGO4:

```
Start Page x Laboratorio7.java x Soldado.java x VideoJuego4.java x
Source History
1  /*
2  | Clase VideoJuego4
3  | */
4  package laboratorio7;
5
6  import java.util.*;
7  public class VideoJuego4 {
8  |
9  |     private Soldado[][] tablero; // Tablero bidimensional de soldados
10 |     public ArrayList<Soldado> ejercito1; // Lista de soldados del Ejército 1
11 |     public ArrayList<Soldado> ejercito2; // Lista de soldados del Ejército 2
12 |     // Constructor: Inicializa el tablero y los ejércitos
13 |     public VideoJuego4() {
14 |         this.tablero=new Soldado[10][10]; // Inicializa el tablero bidimensional
15 |         this.ejercito1=new ArrayList<Soldado>(); // Inicializa la lista del Ejército 1
16 |         this.ejercito2=new ArrayList<Soldado>(); // Inicializa la lista del Ejército 2
17 |         inicializarEjercito(1, ejercito1); // Inicializa los soldados para el Ejército 1
18 |         inicializarEjercito(2, ejercito2); // Inicializa los soldados para el Ejército 2
19 |     }
20 |     // Método para inicializar los soldados en el tablero y añadirlos a un ejército
21 |     private void inicializarEjercito(int numEjercito, ArrayList<Soldado> ejercito){
22 |         int cantidadSoldados=(int) (Math.random()*10+1); // Genera una cantidad aleatoria de soldados entre 1 y 10
23 |         for(int i=0;i<cantidadSoldados;i++){
24 |             String nombre="Soldado"+i+"X"+numEjercito; // Genera un nombre único para cada soldado
25 |             int nivelVida=(int) (Math.random()*5+1); // Asigna un nivel de vida aleatorio entre 1 y 5
26 |             int fila, columna;
27 |             // Busca una posición vacía en el tablero para colocar al soldado
28 |             do{
29 |                 fila=(int) (Math.random()*10); // Fila aleatoria
30 |                 columna=(int) (Math.random()*10); // Columna aleatoria
31 |             }while(tablero[fila][columna]!=null); // Repite mientras la posición no esté vacía
32 |
33 |             Soldado soldado=new Soldado(nombre, nivelVida, fila, columna); // Crea el soldado con los atributos generados
34 |             tablero[fila][columna]=soldado; // Coloca el soldado en el tablero
35 |             ejercito.add(soldado); // Añade el soldado al ejército correspondiente
36 |         }
37 |     }
38 |     // Método para mostrar el tablero con los soldados posicionados
39 |     public void mostrarTablero() {
40 |         for(int i=0;i<tablero.length;i++){
41 |             for(int j=0;j<tablero[i].length;j++){
42 |                 Soldado soldado=tablero[i][j]; // Obtiene el soldado en la posición (i,j)
43 |                 if(soldado==null){
44 |                     System.out.print("| _____ "); // Imprime una casilla vacía
45 |                 }else{
46 |                     System.out.print("| "+soldado.getNombre()+" "); // Imprime el nombre del soldado en la casilla
47 |                 }
48 |             }
49 |             System.out.println("|"); // Nueva línea después de cada fila
50 |         }
51 |         System.out.println();
52 |     }
53 | }
```

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

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

	<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: 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.

LABORATORIO7

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

```

run:
| _____ | _____ | _____ | _____ | _____ | Soldado4X2 | _____ | Soldado3X1 | _____ | _____ |
| _____ | Soldado0X1 | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | _____ | Soldado2X1 | _____ | _____ | _____ | Soldado5X2 | _____ | _____ | _____ |
| _____ | _____ | _____ | _____ | _____ | _____ | Soldado4X1 | _____ | _____ | _____ |
| _____ | _____ | Soldado3X2 | _____ | _____ | Soldado2X2 | _____ | Soldado1X2 | _____ | _____ |
| _____ | _____ | _____ | Soldado0X2 | Soldado1X1 | _____ | _____ | _____ | _____ | _____ |
| _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | Soldado5X1 |
| _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |

```

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

Soldado con mayor nivel de vida del ejercito 1 : Soldado: Soldado3X1, Vida: 5, Posicion: (1, 8)

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

Soldado con mayor nivel de vida del ejercito 2 : Soldado: Soldado1X2, Vida: 4, Posicion: (6, 8)

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

Promedio del nivel de vida del ejercito 1 : 1.8333333333333333

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: 1, Posicion: (2, 2)
Soldado: Soldado1X1, Vida: 2, Posicion: (7, 5)
Soldado: Soldado2X1, Vida: 1, Posicion: (4, 3)
Soldado: Soldado3X1, Vida: 5, Posicion: (1, 8)
Soldado: Soldado4X1, Vida: 1, Posicion: (5, 7)
Soldado: Soldado5X1, Vida: 1, Posicion: (8, 10)
```

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



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

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: Soldado3X1, Vida: 5, Posicion: (1, 8)  
Soldado: Soldado1X1, Vida: 2, Posicion: (7, 5)  
Soldado: Soldado0X1, Vida: 1, Posicion: (2, 2)  
Soldado: Soldado2X1, Vida: 1, Posicion: (4, 3)  
Soldado: Soldado4X1, Vida: 1, Posicion: (5, 7)  
Soldado: Soldado5X1, Vida: 1, Posicion: (8, 10)  
Ranking de soldados ejercito 1 (Seleccion):  
Soldado: Soldado3X1, Vida: 5, Posicion: (1, 8)  
Soldado: Soldado1X1, Vida: 2, Posicion: (7, 5)  
Soldado: Soldado2X1, Vida: 1, Posicion: (4, 3)  
Soldado: Soldado0X1, Vida: 1, Posicion: (2, 2)  
Soldado: Soldado4X1, Vida: 1, Posicion: (5, 7)  
Soldado: Soldado5X1, Vida: 1, Posicion: (8, 10)
```

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: Soldado1X2, Vida: 4, Posicion: (6, 8)  
Soldado: Soldado4X2, Vida: 4, Posicion: (1, 6)  
Soldado: Soldado5X2, Vida: 4, Posicion: (4, 7)  
Soldado: Soldado3X2, Vida: 2, Posicion: (6, 3)  
Soldado: Soldado0X2, Vida: 1, Posicion: (7, 4)  
Soldado: Soldado2X2, Vida: 1, Posicion: (6, 6)  
Ranking de soldados ejercito 2 (Seleccion):  
Soldado: Soldado1X2, Vida: 4, Posicion: (6, 8)  
Soldado: Soldado4X2, Vida: 4, Posicion: (1, 6)  
Soldado: Soldado5X2, Vida: 4, Posicion: (4, 7)  
Soldado: Soldado3X2, Vida: 2, Posicion: (6, 3)  
Soldado: Soldado0X2, Vida: 1, Posicion: (7, 4)  
Soldado: Soldado2X2, Vida: 1, Posicion: (6, 6)
```

	<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.

```

Hubo un EMPATE de ejercitos, ambos con 6 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: 4 minutes 0 seconds)

```

11. Ejecución completa.

```

Output - Laboratorio7 (run) x
run:
| _____ | _____ | _____ | _____ | _____ | Soldado4X2 | _____ | Soldado3X1 | _____ | _____ |
| _____ | Soldado0X1 | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | _____ | Soldado2X1 | _____ | _____ | _____ | Soldado5X2 | _____ | _____ | _____ |
| _____ | _____ | _____ | _____ | _____ | _____ | Soldado4X1 | _____ | _____ | _____ |
| _____ | _____ | Soldado3X2 | _____ | _____ | Soldado2X2 | _____ | Soldado1X2 | _____ | _____ |
| _____ | _____ | _____ | Soldado0X2 | Soldado1X1 | _____ | _____ | _____ | _____ | _____ |
| _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | Soldado5X1 |
| _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |

Soldado con mayor nivel de vida del ejercito 1 : Soldado: Soldado3X1, Vida: 5, Posicion: (1, 8)

Soldado con mayor nivel de vida del ejercito 2 : Soldado: Soldado1X2, Vida: 4, Posicion: (6, 8)

Promedio del nivel de vida del ejercito 1 : 1.8333333333333333

Promedio del nivel de vida del ejercito 2 : 2.6666666666666665

Soldados del ejercito 1 en el orden de creacion:
Soldado: Soldado0X1, Vida: 1, Posicion: (2, 2)
Soldado: Soldado1X1, Vida: 2, Posicion: (7, 5)
Soldado: Soldado2X1, Vida: 1, Posicion: (4, 3)
Soldado: Soldado3X1, Vida: 5, Posicion: (1, 8)
Soldado: Soldado4X1, Vida: 1, Posicion: (5, 7)
Soldado: Soldado5X1, Vida: 1, Posicion: (8, 10)

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

Ranking de soldados ejercito 1 (Burbuja):
Soldado: Soldado3X1, Vida: 5, Posicion: (1, 8)
Soldado: Soldado1X1, Vida: 2, Posicion: (7, 5)
Soldado: Soldado0X1, Vida: 1, Posicion: (2, 2)
Soldado: Soldado2X1, Vida: 1, Posicion: (4, 3)
Soldado: Soldado4X1, Vida: 1, Posicion: (5, 7)
Soldado: Soldado5X1, Vida: 1, Posicion: (8, 10)
Ranking de soldados ejercito 1 (Seleccion):
Soldado: Soldado3X1, Vida: 5, Posicion: (1, 8)
Soldado: Soldado1X1, Vida: 2, Posicion: (7, 5)
Soldado: Soldado2X1, Vida: 1, Posicion: (4, 3)
Soldado: Soldado0X1, Vida: 1, Posicion: (2, 2)
Soldado: Soldado4X1, Vida: 1, Posicion: (5, 7)
Soldado: Soldado5X1, Vida: 1, Posicion: (8, 10)

```

Ranking de soldados ejercito 2 (Burbuja):

Soldado: Soldado1X2, Vida: 4, Posicion: (6, 8)

Soldado: Soldado4X2, Vida: 4, Posicion: (1, 6)

Soldado: Soldado5X2, Vida: 4, Posicion: (4, 7)

Soldado: Soldado3X2, Vida: 2, Posicion: (6, 3)

Soldado: Soldado0X2, Vida: 1, Posicion: (7, 4)

Soldado: Soldado2X2, Vida: 1, Posicion: (6, 6)

Ranking de soldados ejercito 2 (Seleccion):

Soldado: Soldado1X2, Vida: 4, Posicion: (6, 8)

Soldado: Soldado4X2, Vida: 4, Posicion: (1, 6)

Soldado: Soldado5X2, Vida: 4, Posicion: (4, 7)

Soldado: Soldado3X2, Vida: 2, Posicion: (6, 3)

Soldado: Soldado0X2, Vida: 1, Posicion: (7, 4)

Soldado: Soldado2X2, Vida: 1, Posicion: (6, 6)



Hubo un EMPATE de ejercitos, ambos con 6 soldados...

Desea generar una nueva batalla?(s/n)

n

Gracias por jugar!

BUILD SUCCESSFUL (total time: 4 minutes 0 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;">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>

MIS COMMITS:

PRIMERA VERSION:

```

user@DESKTOP-CS7PIHP MINGW64 ~ (master)
$ cd LABORATORIOS_FP2

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git add LABORATORIO_07

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git commit -m "Haciendo el programa iterativo"
[main 903cdc2] Haciendo el programa iterativo
3 files changed, 218 insertions(+)
create mode 100644 LABORATORIO_07/Laboratorio7.java
create mode 100644 LABORATORIO_07/Soldado.java
create mode 100644 LABORATORIO_07/VideoJuego4.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), 3.26 KiB | 3.26 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
52a1042..903cdc2  main -> main

```

Se agregaron los archivos Laboratorio7.java, Soldado.java y VideoJuego4.java al repositorio de GitHub.

SEGUNDA VERSION:

```

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git add LABORATORIO_07

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git commit -m "Version final: combinando arreglos estandar con arraylist "
[main 54a514e] Version final: combinando arreglos estandar con arraylist
2 files changed, 9 insertions(+), 23 deletions(-)



user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 623 bytes | 623.00 KiB/s, done.
Total 5 (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
903cdc2..54a514e  main -> main

```

Se actualizaron los archivos Laboratorio7.java, Soldado.java y VideoJuego4.java y se subieron a GitHub.

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;">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: 16</p>

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:

En esta práctica, se implementó un sistema de soldados en un tablero bidimensional utilizando arreglos estándar y ArrayList, lo que permitió simular una batalla entre dos ejércitos. La organización en clases y el uso de algoritmos de ordenamiento facilitaron la gestión y clasificación de los soldados. Las pruebas realizadas confirmaron el funcionamiento correcto del programa y demostraron la importancia de elegir adecuadamente las estructuras de datos para optimizar el rendimiento y la precisión en la programación.

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.

- **Análisis del problema:** Se revisaron los requisitos de la práctica, comprendiendo la necesidad de implementar un tablero bidimensional y trabajar con soldados distribuidos aleatoriamente combinando arreglos estándar y arraylist.
- **Diseño de clases:** Se definieron las clases Soldado y VideoJuego4 para estructurar los objetos soldados y gestionar el tablero y las operaciones requeridas.
- **Implementación del código:** Se programaron las funcionalidades clave, incluyendo la creación aleatoria de soldados, la muestra de los soldados generados y los métodos de ordenamiento por burbuja y selección.
- **Pruebas y validación:** Se realizaron pruebas ejecutando el juego con diferentes tamaños de tablero y cantidades de soldados, verificando el correcto posicionamiento de los soldados y los cálculos de vida.
- **Depuración:** Se corrigieron errores encontrados durante las pruebas, optimizando el código y ajustando el manejo de posiciones vacías en el tablero.
- **Documentación:** Finalmente, se documentó el código y las versiones fueron subidas a un repositorio en GitHub para su revisión y control de versiones.

	<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: 17</p>

REFERENCIAS Y BIBLIOGRAFÍA

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

M. Aedo López, Práctica de Laboratorio 7: Combinando Arreglos Estándar y Arraylist, Universidad Nacional de San Agustín, 2023.

<https://github.com/ChipanaGerman/LABORATORIOS-FP2>