
	<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>Arreglos Bidimensionales de Objetos</i>				
<b>NÚMERO DE PRÁCTICA:</b>	<i>05</i>	<b>AÑO LECTIVO:</b>	<i>2024</i>	<b>NRO. SEMESTRE:</b>	<i>2</i>
<b>FECHA DE PRESENTACIÓN</b>	<i>18/10/2024</i>	<b>HORA DE PRESENTACIÓN</b>	<i>17/20/00</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.</li> <li>2. Usted deberá crear las dos clases Soldado.java y VideoJuego2.java. Puede reutilizar lo desarrollado en Laboratorio 3 y 4.</li> <li>3. Del Soldado nos importa el nombre, nivel de vida, fila y columna.</li> <li>4. El juego se desarrollará en el mismo tablero de los laboratorios anteriores. Pero ahora el tablero debe ser un arreglo bidimensional de objetos.</li> <li>5. Inicializar el tablero con n soldados aleatorios entre 1 y 10. Cada soldado tendrá un nombre autogenerado: Soldado0, Soldado1, etc., un valor de nivel de vida autogenerado aleatoriamente [1..5], la fila y columna también autogenerados aleatoriamente (verificar que no puede haber 2 soldados en el mismo cuadrado). Se debe mostrar el tablero con todos los soldados creados (usar caracteres como   _ y otros). Además, mostrar los datos del Soldado con mayor nivel de vida, el promedio de nivel de vida de todos los soldados creados, el nivel de vida de todo el ejército, los datos de todos los soldados en el orden que fueron creados y un ranking de poder de todos los soldados creados, del que tiene más nivel de vida al que tiene menos (usar al menos 2 algoritmos de ordenamiento).</li> </ol>

**MAIN:**

```
1  /*
2  |   Autor: Chipana Jeronimo German Arturo
3  |   Proposito: Ejercicio 01
4  |   */
5  package ejercicio01lab05;
6
7
8  public class Ejercicio01Lab05 {
9
10 |     public static void main(String[] args) {
11 |         VideoJuego2 juego=new VideoJuego2();
12 |         juego.mostrarTablero();
13 |         juego.mostrarSoldadoMayorVida();
14 |         juego.mostrarPromedioNivelVida();
15 |         juego.mostrarVidaTotalEjercito();
16 |         juego.mostrarSoldados();
17 |         juego.mostrarRankingSoldados();
18 |     }
19 | }
20
```

**CLASE SOLDADO:**

```
1  /*
2  |   Clase Soldado
3  |   */
4  package ejercicio01lab05;
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 accesoros
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 |
33 |   public String toString(){
34 |       return "Soldado: "+nombre+" Vida: "+nivelVida+", Posicion: ("+fila+", "+columna+")";
35 |   }
36 | }
37
```

**CLASE VIDEOJUEGO2:**

```
1  /*
2  Clase VideoJuego2
3  */
4  package ejercicio01lab05;
5
6
7  public class VideoJuego2{
8      private Soldado[][] tablero; // Tablero bidimensional de soldados
9      private Soldado[] soldados; // Arreglo de soldados
10     private int cantidadSoldados; // Cantidad de soldados
11     // Constructor
12     public VideoJuego2(){
13         this.tablero=new Soldado[10][10];
14         this.cantidadSoldados=(int) (Math.random()*10+1); // Entre 1 y 10 soldados
15         this.soldados=new Soldado[cantidadSoldados];
16         inicializarSoldados();
17     }
18     // Método para inicializar los soldados en el tablero
19     private void inicializarSoldados(){
20         for(int i=0;i<cantidadSoldados;i++){
21             String nombre="Soldado"+i;
22             int nivelVida=(int) (Math.random()*5+1); // Nivel de vida entre 1 y 5
23             int fila, columna;
24             // Generar una posición única para el soldado
25             do{
26                 fila=(int) (Math.random()*10);
27                 columna=(int) (Math.random()*10);
28             }while(tablero[fila][columna]!=null); // Asegurarse de que el espacio esté ocupado
29
30             Soldado soldado=new Soldado(nombre, nivelVida, fila, columna);
31             tablero[fila][columna]=soldado; // Asignar el soldado al tablero
32             soldados[i]=soldado; // Guardar soldado en el arreglo
33         }
34     }
35     // Método para mostrar el tablero
36     public void mostrarTablero(){
37         for(int i=0;i<10;i++){
38             for(int j=0;j<10;j++){
39                 if(tablero[i][j]==null){
40                     System.out.print("| _____ "); // Posición vacía
41                 }else{
42                     System.out.print("| "+tablero[i][j].getNombre()+" "); // Posición con un soldado
43                 }
44             }
45             System.out.println("|");
46         }
47     }
48 }
```

```

48 // Método para mostrar el soldado con mayor nivel de vida
49 public void mostrarSoldadoMayorVida(){
50     Soldado mayorVida=soldados[0];
51     for(Soldado soldado : soldados){
52         if(soldado.getNivelVida()>mayorVida.getNivelVida()){
53             mayorVida=soldado;
54         }
55     }
56     System.out.println("Soldado con mayor nivel de vida: "+mayorVida);
57 }
58 // Método para calcular el promedio de nivel de vida
59 public void mostrarPromedioNivelVida(){
60     int sumaVida=0;
61     for(Soldado soldado : soldados){
62         sumaVida+=soldado.getNivelVida();
63     }
64     double promedio=(double)sumaVida/cantidadSoldados;
65     System.out.println("Promedio del nivel de vida: "+promedio);
66 }
67 // Método para mostrar el nivel de vida total del ejercito
68 public void mostrarVidaTotalEjercito(){
69     int vidaTotal=0;
70     for(Soldado soldado : soldados){
71         vidaTotal+=soldado.getNivelVida();
72     }
73     System.out.println("Nivel de vida total del ejercito: "+vidaTotal);
74 }
75 // Método para mostrar todos los soldados en el orden en que fueron creados
76 public void mostrarSoldados(){
77     System.out.println("Soldados en el orden de creacion:");
78     for(Soldado soldado : soldados){
79         System.out.println(soldado);
80     }
81 }
82 // Método para mostrar el ranking de los soldados
83 public void mostrarRankingSoldados(){
84     // Usar método de burbuja
85     Soldado[] soldadosBurbuja=soldados.clone();
86     burbujaOrdenar(soldadosBurbuja);
87     System.out.println("Ranking de soldados (Burbuja): ");
88     for(Soldado soldado : soldadosBurbuja){
89         System.out.println(soldado);
90     }
91     // Usar método de selección
92     Soldado[] soldadosSeleccion=soldados.clone();
93     seleccionOrdenar(soldadosSeleccion);
94     System.out.println("Ranking de soldados (Seleccion): ");
95     for(Soldado soldado : soldadosSeleccion){
96         System.out.println(soldado);
97     }
98 }

```

```

99      // Método de ordenamiento por burbuja
100     private void burbujaOrdenar(Soldado[] soldados){
101         for(int i=0;i<soldados.length-1;i++){
102             for(int j=0;j<soldados.length-1-i;j++){
103                 if(soldados[j].getNivelVida()<soldados[j+1].getNivelVida()){
104                     Soldado temp=soldados[j];
105                     soldados[j]=soldados[j+1];
106                     soldados[j+1]=temp;
107                 }
108             }
109         }
110     }
111     // Método de ordenamiento por selección
112     private void seleccionOrdenar(Soldado[] soldados){
113         for(int i=0;i<soldados.length-1;i++){
114             int maxIndex=i;
115             for(int j=i+1;j<soldados.length;j++){
116                 if(soldados[j].getNivelVida()>soldados[maxIndex].getNivelVida()){
117                     maxIndex=j;
118                 }
119             }
120             Soldado temp=soldados[i];
121             soldados[i]=soldados[maxIndex];
122             soldados[maxIndex]=temp;
123         }
124     }
125 }

```

## II. PRUEBAS

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

Utilicé diferentes ejecuciones con números de soldados aleatorios entre 1 y 10, generados mediante `Math.random()`. Verifiqué que los soldados se colocaran correctamente en el tablero y que sus niveles de vida se generaran dentro del rango de 1 a 5.

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

Esperaba que los soldados se distribuyeran aleatoriamente en el tablero de 10x10, sin repetir posiciones. También esperaba que los métodos identificaran correctamente al soldado con mayor nivel de vida, calcularan el promedio y total de vida de los soldados, y mostraran el ranking ordenado correctamente de mayor a menor nivel de vida usando dos algoritmos.

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

Los soldados fueron distribuidos aleatoriamente y sin repetir posiciones en el tablero. El soldado con mayor nivel de vida fue correctamente identificado, el promedio y total de vida fueron calculados de manera precisa, y el ranking de soldados mostró el orden correcto en ambas implementaciones de ordenamiento (burbuja y selección). Los resultados fueron consistentes con lo esperado.

Nivel de vida total del ejercito: 31

**5. Se mostró a todos los soldados en el orden de creación con sus datos correspondientes.**



Soldados en el orden de creacion:

Soldado: Soldado0, Vida: 5, Posicion: (1, 9)  
Soldado: Soldado1, Vida: 5, Posicion: (6, 6)  
Soldado: Soldado2, Vida: 1, Posicion: (6, 9)  
Soldado: Soldado3, Vida: 2, Posicion: (3, 1)  
Soldado: Soldado4, Vida: 1, Posicion: (0, 3)  
Soldado: Soldado5, Vida: 5, Posicion: (1, 3)  
Soldado: Soldado6, Vida: 2, Posicion: (5, 3)  
Soldado: Soldado7, Vida: 3, Posicion: (9, 3)  
Soldado: Soldado8, Vida: 4, Posicion: (1, 5)  
Soldado: Soldado9, Vida: 3, Posicion: (3, 8)

**6. Se mostró a los soldados ordenados por el nivel de vida (mayor a menor) por el ordenamiento burbuja.**

Ranking de soldados (Burbuja):

Soldado: Soldado0, Vida: 5, Posicion: (1, 9)  
Soldado: Soldado1, Vida: 5, Posicion: (6, 6)  
Soldado: Soldado5, Vida: 5, Posicion: (1, 3)  
Soldado: Soldado8, Vida: 4, Posicion: (1, 5)  
Soldado: Soldado7, Vida: 3, Posicion: (9, 3)  
Soldado: Soldado9, Vida: 3, Posicion: (3, 8)  
Soldado: Soldado3, Vida: 2, Posicion: (3, 1)  
Soldado: Soldado6, Vida: 2, Posicion: (5, 3)  
Soldado: Soldado2, Vida: 1, Posicion: (6, 9)  
Soldado: Soldado4, Vida: 1, Posicion: (0, 3)

	<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 style="text-align: center;">Código: GUIA-PRLE-001</p>	<p style="text-align: right;">Página: 8</p>

**7. Se mostró a los soldados ordenados por el nivel de vida (mayor a menor) por el ordenamiento selección.**

```

Ranking de soldados (Selección):
Soldado: Soldado0, Vida: 5, Posición: (1, 9)
Soldado: Soldado1, Vida: 5, Posición: (6, 6)
Soldado: Soldado5, Vida: 5, Posición: (1, 3)
Soldado: Soldado8, Vida: 4, Posición: (1, 5)
Soldado: Soldado7, Vida: 3, Posición: (9, 3)
Soldado: Soldado9, Vida: 3, Posición: (3, 8)
Soldado: Soldado6, Vida: 2, Posición: (5, 3)
Soldado: Soldado3, Vida: 2, Posición: (3, 1)
Soldado: Soldado4, Vida: 1, Posición: (0, 3)
Soldado: Soldado2, Vida: 1, Posición: (6, 9)
BUILD SUCCESSFUL (total time: 0 seconds)

```





## 8.SALIDA COMPLETA.

```

Output - Ejercicio01Lab05 (run) x
run:
| _____ | _____ | _____ | Soldado4 | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | _____ | _____ | Soldado5 | _____ | Soldado8 | _____ | _____ | _____ | Soldado0 |
| _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | Soldado3 | _____ | _____ | _____ | _____ | _____ | _____ | Soldado9 | _____ |
| _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | _____ | _____ | Soldado6 | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | _____ | _____ | _____ | _____ | _____ | Soldado1 | _____ | _____ | Soldado2 |
| _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ | _____ |
| _____ | _____ | _____ | Soldado7 | _____ | _____ | _____ | _____ | _____ | _____ |

Soldado con mayor nivel de vida: Soldado: Soldado0, Vida: 5, Posicion: (1, 9)
Promedio del nivel de vida: 3.1
Nivel de vida total del ejercito: 31
Soldados en el orden de creacion:
Soldado: Soldado0, Vida: 5, Posicion: (1, 9)
Soldado: Soldado1, Vida: 5, Posicion: (6, 6)
Soldado: Soldado2, Vida: 1, Posicion: (6, 9)
Soldado: Soldado3, Vida: 2, Posicion: (3, 1)
Soldado: Soldado4, Vida: 1, Posicion: (0, 3)
Soldado: Soldado5, Vida: 5, Posicion: (1, 3)
Soldado: Soldado6, Vida: 2, Posicion: (5, 3)
Soldado: Soldado7, Vida: 3, Posicion: (9, 3)
Soldado: Soldado8, Vida: 4, Posicion: (1, 5)
Soldado: Soldado9, Vida: 3, Posicion: (3, 8)
Ranking de soldados (Burbuja):
Soldado: Soldado0, Vida: 5, Posicion: (1, 9)
Soldado: Soldado1, Vida: 5, Posicion: (6, 6)
Soldado: Soldado5, Vida: 5, Posicion: (1, 3)
Soldado: Soldado8, Vida: 4, Posicion: (1, 5)
Soldado: Soldado7, Vida: 3, Posicion: (9, 3)
Soldado: Soldado9, Vida: 3, Posicion: (3, 8)
Soldado: Soldado3, Vida: 2, Posicion: (3, 1)
Soldado: Soldado6, Vida: 2, Posicion: (5, 3)
Soldado: Soldado2, Vida: 1, Posicion: (6, 9)
Soldado: Soldado4, Vida: 1, Posicion: (0, 3)
Ranking de soldados (Seleccion):
Soldado: Soldado0, Vida: 5, Posicion: (1, 9)
Soldado: Soldado1, Vida: 5, Posicion: (6, 6)
Soldado: Soldado5, Vida: 5, Posicion: (1, 3)
Soldado: Soldado8, Vida: 4, Posicion: (1, 5)
Soldado: Soldado7, Vida: 3, Posicion: (9, 3)
Soldado: Soldado9, Vida: 3, Posicion: (3, 8)
Soldado: Soldado6, Vida: 2, Posicion: (5, 3)
Soldado: Soldado3, Vida: 2, Posicion: (3, 1)
Soldado: Soldado4, Vida: 1, Posicion: (0, 3)
Soldado: Soldado2, Vida: 1, Posicion: (6, 9)
BUILD SUCCESSFUL (total time: 0 seconds)

```

	<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> 10</p>

## MIS COMMITS:

### PRIMERA VERSION:

```

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git add LABORATORIO_05

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git commit -m "Primera Version Lab 05"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git push origin main
fatal: unable to access 'https://github.com/ChipanaGerman/LABORATORIOS-FP2.git/':
OpenSSL SSL_read: SSL_ERROR_SYSCALL, errno 0

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), 1.67 KiB | 1.67 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
  795aa7e..963c001  main -> main

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

```

### SEGUNDA VERSION:

```

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git add LABORATORIO_05



user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git commit -m "Version oficial Lab 05"
[main 58bc650] Version oficial Lab 05
 3 files changed, 77 insertions(+), 1 deletion(-)

user@DESKTOP-CS7PIHP MINGW64 ~/LABORATORIOS_FP2 (main)
$ git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.73 KiB | 1.73 MiB/s, done.
Total 6 (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
  963c001..58bc650  main -> main

```

### 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 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> 12

*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 sesión de laboratorio, se comprendió y aplicó el manejo de arreglos bidimensionales de objetos en Java, lo que permitió almacenar y manipular datos de manera eficiente. La correcta implementación de métodos y la solución de los ejercicios fortalecieron las habilidades en programación orientada a objetos, así como en la depuración y documentación del código

## 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.*

- Se analizaron los ejercicios propuestos y se comprendió la lógica de los problemas.
- Se implementó el código inicial utilizando arreglos bidimensionales de objetos para las clases solicitadas.
- Se realizaron pruebas para verificar el correcto funcionamiento del programa, introduciendo diferentes valores de entrada.
- Se corrigieron los errores encontrados durante las pruebas.
- Finalmente, se documentó el código y se realizaron los commits correspondientes en GitHub

## REFERENCIAS Y BIBLIOGRAFÍA

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

M. Aedo López, Práctica de Laboratorio 5: Arreglos Bidimensionales de Objetos, Universidad Nacional de San Agustín, 2023.

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