

Informe de Laboratorio 05

Tema: Arreglos Bidimensionales de Objetos

Nota

Estudiante	Escuela	Asignatura
Christian Mestas Zegarra cmestasz@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Fundamentos de la Programación 2 Semestre: II Código: 1701213

Laboratorio	Tema	Duración
05	Arreglos Bidimensionales de Objetos	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 11 Octubre 2023	Al 16 Octubre 2023

1. Tarea

- **Actividad 1:** Cree un Proyecto llamado Laboratorio5.
- **Actividad 2:** Usted deberá crear las dos clases Soldado.java y VideoJuego2.java. Puede reutilizar lo desarrollado en Laboratorio 3 y 4.
- **Actividad 3:** Del Soldado nos importa el nombre, puntos de vida, fila y columna (posición en el tablero).
- **Actividad 4:** El juego se desarrollará en el mismo tablero de los laboratorios anteriores. Pero ahora el tablero debe ser un arreglo bidimensional de objetos.
- **Actividad 5:** Inicializar el tablero con n soldados aleatorios entre 1 y 10. Cada soldado tendrá un nombre autogenerado: Soldado0, Soldado1, 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 (usar caracteres como | _ y otros). Además de los datos del Soldado con mayor vida, el promedio de puntos 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).

2. Equipos, materiales y temas utilizados

- Sistema Operativo Microsoft Windows 10 Pro 64 bits
- Visual Studio Code 1.82.2
- Java Development Kit 17.0.1
- Git 2.41.0.windows.1
- Windows PowerShell 5.1.19041.3031
- Cuenta en GitHub con el correo institucional.
- Arreglos bidimensionales de objetos.
- Ordenamientos burbuja y por selección.

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/cmestasz/fp2-23b.git>
- URL para el laboratorio 05 en el Repositorio GitHub.
- <https://github.com/cmestasz/fp2-23b/tree/main/fase01/lab05>

4. Actividades con el repositorio GitHub

Creando plantillas

```
$ mkdir lab05  
$ cd lab05  
$ code Soldado.java  
$ code VideoJuego2.java
```

Primer Commit / Plantillas

```
$ git add .  
$ git commit -m "Plantillas de archivos necesarios para resolver las actividades del  
laboratorio"  
$ git push
```

Showing 2 changed files with 32 additions and 0 deletions.

- base01/lab05
 - Soldado.java
 - VideoJuego2.java

```

1 + public class Soldado {
2 +     private String nombre;
3 +     private int vida;
4 +
5 +     public Soldado(String nombre, int vida) {
6 +         setNombre(nombre);
7 +         setVida(vida);
8 +     }
9 +
10 +    public void setNombre(String nombre) {
11 +        this.nombre = nombre;
12 +    }
13 +
14 +    public void setVida(int vida) {
15 +        this.vida = vida;
16 +    }
17 +
18 +    public String getNombre() {
19 +        return nombre;
20 +    }
21 +
22 +    public int getVida() {
23 +        return vida;
24 +    }
25 +
26 +    public String toString() {
27 +        return "Nombre: " + nombre + ", Vida: " + vida;
28 +    }
29 + }

```

3
base01/lab05/VideoJuego2.java

```

1 + public class VideoJuego2 {
2 +
3 + }

```

Primer Commit.

Actualizando VideoJuego2.java

```
$ code VideoJuego2.java
```

Segundo - Decimo Commit / DemoBatalla.java

```

$ git add .
$ git commit -m "Main y plantillas de metodos a utilizar"
$ code DemoBatalla.java
$ git add DemoBatalla.java
$ git commit -m "Metodo inicializarSoldados()"
$ code DemoBatalla.java
$ git add DemoBatalla.java
$ git commit -m "Metodo imprimirTablero(), incompleto"
$ code DemoBatalla.java
$ git add DemoBatalla.java
$ git commit -m "Metodo imprimirTablero() y auxiliares"
$ code DemoBatalla.java
$ git add DemoBatalla.java
$ git commit -m "Main, inicializarSoldados() y demas modificados"
$ code DemoBatalla.java
$ git add DemoBatalla.java
$ git commit -m "Metodos soldadoMayorVida(), promedioPuntosVida(), sumaPuntosVida()"
$ code DemoBatalla.java
$ git add DemoBatalla.java
$ git commit -m "Metodo ordenarSoldadosBurbuja()"
$ code DemoBatalla.java
$ git add DemoBatalla.java
$ git commit -m "Metodo ordenarSoldadosSeleccion()"
$ code DemoBatalla.java
$ git add DemoBatalla.java
$ git commit -m "Metodo imprimirSoldados()"
$ git push

```

Showing 1 changed file with 47 additions and 1 deletion.

<pre> 1 public class videojuego2 { 2 + 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 </pre>	<pre> 1 public class videojuego2 { 2 + 3 public static void main(String[] args) { 4 + 5 Soldado[][] soldados = new Soldado[10][10]; 6 + 7 inicializarSoldados(soldados); 8 + 9 imprimirTablero(soldados); 10 + 11 System.out.println("Soldado con mayor vida: " + soldadoMayorVida(soldados)); 12 + 13 System.out.println("Promedio de puntos de vida: " + promedioPuntosVida(soldados)); 14 + 15 System.out.println("Nivel de vida del ejercito: " + sumaPuntosVida(soldados)); 16 + 17 Soldado[][] soldados2 = new Soldado[10][10]; 18 + 19 Soldado[][] soldados3 = new Soldado[10][10]; 20 + 21 System.arraycopy(soldados, 0, soldados2, 0, soldados.length); 22 + 23 System.arraycopy(soldados, 0, soldados3, 0, soldados.length); 24 + 25 ordenarSoldadosBurbuja(soldados2); 26 + 27 imprimirSoldados(soldados2); 28 + 29 ordenarSoldadosSelecion(soldados3); 30 + 31 imprimirSoldados(soldados3); 32 + 33 } 34 + 35 private static void inicializarSoldados(Soldado[][] soldados) { 36 + 37 } 38 + 39 private static void imprimirTablero(Soldado[][] soldados) { 40 + 41 } 42 + 43 private static Soldado soldadoMayorVida(Soldado[][] soldados) { 44 + 45 } 46 + 47 private static Soldado promedioPuntosVida(Soldado[][] soldados) { 48 + 49 } 50 + 51 private static Soldado sumaPuntosVida(Soldado[][] soldados) { 52 + 53 } 54 + 55 private static void ordenarSoldadosBurbuja(Soldado[][] soldados2) { 56 + 57 } 58 + 59 private static void ordenarSoldadosSelecion(Soldado[][] soldados3) { 60 + 61 } 62 + 63 private static void imprimirSoldados(Soldado[][] soldados) { 64 + 65 } 66 + 67 public static int charToInt(char c) { 68 + 69 return c - 'A' + 1; 70 + 71 } 72 + 73 } 74 + 75 } </pre>
---	--

Segundo Commit.

Showing 1 changed file with 23 additions and 3 deletions.

<pre> 19 private static void inicializarSoldados(Soldado[][] soldados) { 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 </pre>	<pre> 22 private static void inicializarSoldados(Soldado[][] soldados) { 23 + 24 Random r = new Random(); 25 + 26 int cantidad = r.nextInt(10) + 1; 27 + 28 for (int i = 0; i < cantidad; i++) { 29 + 30 String nombre = "Soldado" + i; 31 + 32 int vida = r.nextInt(5) + 1; 33 + 34 int fila, columna; 35 + 36 do { 37 + 38 fila = r.nextInt(10); 39 + 40 columna = r.nextInt(10); 41 + 42 } while (soldados[fila][columna] != null); 43 + 44 soldados[fila][columna] = new Soldado(nombre, vida); 45 + 46 } 47 + 48 } 49 + 50 } 51 + 52 } </pre>
---	---

Tercer Commit.

Showing 1 changed file with 14 additions and 0 deletions.

<pre> 38 private static void imprimirTablero(Soldado[][] soldados) { 39 40 41 42 43 44 45 46 47 48 49 50 51 52 </pre>	<pre> 38 private static void imprimirTablero(Soldado[][] soldados) { 39 + 40 System.out.print("\t"); 41 + 42 for (int i = 0; i < soldados[0].length; i++) 43 + 44 System.out.print(" " + infoChar(i + 1) + " "); 45 + 46 System.out.println(); 47 + 48 for (int i = 0; i < soldados.length; i++) { 49 + 50 System.out.print("\t"); 51 + 52 for (int j = 0; j < soldados[i].length; j++) 53 + 54 System.out.print("..."); 55 + 56 System.out.println(); 57 + 58 System.out.print((i + 1) + "\t"); 59 + 60 for (int j = 0; j < soldados[i].length; j++) 61 + 62 System.out.print(" "); 63 + 64 } 65 + 66 } 67 + 68 } </pre>
---	---

Cuarto Commit.

Showing 1 changed file with 37 additions and 13 deletions.

<pre> 38 private static void imprimirTablero(Soldado[][] soldados) { 39 System.out.println("\n"); 40 for (int i = 0; i < soldados[0].length; i++) 41 System.out.print(" " + Integer.toString(i + 1) + " "); 42 System.out.println(); 43 for (int i = 0; i < soldados.length; i++) { 44 System.out.print("\n"); 45 for (int j = 0; j < soldados[i].length; j++) 46 System.out.print("..."); 47 System.out.println(); 48 System.out.print("(" + i + 1) + "\n"); 49 for (int j = 0; j < soldados[i].length; j++) 50 System.out.print(" " + Integer.toString(j + 1) + " "); 51 System.out.println(); 52 } 53 }</pre>	<pre> 38 private static void imprimirTablero(Soldado[][] soldados) { 39 System.out.print(generarEncabezado(soldados)); 40 String subfilas = generarSubfilas(soldados); 41 for (int i = 0; i < soldados.length; i++) { 42 System.out.print(subfilas); 43 System.out.print(generarSubfilas(soldados, i)); 44 } 45 System.out.print(subfilas); 46 } 47 48 public static String generarEncabezado(Soldado[][] soldados) { 49 String encabezado = "\n"; 50 for (int i = 0; i < soldados.length; i++) 51 encabezado += " " + Integer.toString(i + 1) + " "; 52 encabezado += "\n"; 53 return encabezado; 54 } 55 56 public static String generarSubfilas(Soldado[][] soldados) { 57 String fila = "\n"; 58 for (int i = 0; i < soldados[0].length; i++) 59 fila += "..."; 60 fila += "\n"; 61 return fila; 62 } 63 64 public static String generarSubfilas(Soldado[][] soldados, int f) { 65 String fila = "(" + f + 1) + "\n"; 66 for (int i = 0; i < soldados[f].length; i++) { 67 fila += " "; 68 Soldado soldado = soldados[f][i]; 69 if (soldado != null) 70 fila += soldado.getNombre().charAt(soldado.getNombre().length() - 1); 71 else 72 fila += " "; 73 fila += " "; 74 } 75 fila += "\n"; 76 return fila; 77 }</pre>
--	--

Quinto Commit.

Showing 1 changed file with 27 additions and 23 deletions.

<pre> 7 public static void main(String[] args) { 8 Soldado[][] soldados = new Soldado[10][10]; 9 InicializarSoldados(soldados); 10 imprimirTablero(soldados); 11 System.out.println("Soldado con mayor vida: " + soldadoMayorVida(soldados)); 12 System.out.println("Promedio de puntos de vida: " + promedioPuntosVida(soldados)); 13 System.out.println("Nivel de vida del ejercito: " + sumaPuntosVida(soldados)); 14 Soldado[][] soldados2 = new Soldado[10][10]; 15 Soldado[][] soldados3 = new Soldado[10][10]; 16 System.arraycopy(soldados, 0, soldados2, 0, soldados.length); 17 System.arraycopy(soldados, 0, soldados3, 0, soldados.length); 18 ordenarSoldadosBurbuja(soldados2); 19 imprimirSoldados(soldados2); 20 ordenarSoldadosSeleccion(soldados3); 21 imprimirSoldados(soldados3); 22 } 23 24 private static void inicializarSoldados(Soldado[][] soldados) { 25 Random r = new Random(); 26 int cantidad = r.nextInt(10) + 1; 27 for (int i = 0; i < cantidad; i++) { 28 String nombre = "Soldado" + i; 29 int vida = r.nextInt(5) + 1; 30 int fila, columna; 31 do { 32 fila = r.nextInt(10); 33 columna = r.nextInt(10); 34 } while (soldados[fila][columna] != null); 35 soldados[fila][columna] = new Soldado(nombre, vida); 36 } 37 }</pre>	<pre> 7 public static void main(String[] args) { 8 Random r = new Random(); 9 int cantidad = r.nextInt(10) + 1; 10 Soldado[][] tablero = new Soldado[10][10]; 11 Soldado[] soldados = new Soldado[cantidad]; 12 InicializarSoldados(tablero, soldados); 13 imprimirTablero(tablero); 14 System.out.print("Soldado con mayor vida: %s\n", soldadoMayorVida(soldados)); 15 System.out.print("Promedio de puntos de vida: %f\n", promedioPuntosVida(soldados)); 16 System.out.print("Nivel de vida del ejercito: %d\n", sumaPuntosVida(soldados)); 17 Soldado[] soldados2 = new Soldado[soldados.length]; 18 System.arraycopy(soldados, 0, soldados2, 0, soldados.length); 19 System.arraycopy(soldados, 0, soldados3, 0, soldados.length); 20 System.out.println(); 21 ordenarSoldadosBurbuja(soldados2); 22 imprimirSoldados(soldados2); 23 System.out.println(); 24 ordenarSoldadosSeleccion(soldados3); 25 imprimirSoldados(soldados3); 26 } 27 28 private static void inicializarSoldados(Soldado[][] tablero, Soldado[] soldados) { 29 Random r = new Random(); 30 for (int i = 0; i < soldados.length; i++) { 31 String nombre = "Soldado" + i; 32 int vida = r.nextInt(5) + 1; 33 int fila, columna; 34 do { 35 fila = r.nextInt(10); 36 columna = r.nextInt(10); 37 } while (tablero[fila][columna] != null); 38 Soldado soldado = new Soldado(nombre, vida); 39 tablero[fila][columna] = soldado; 40 soldados[i] = soldado; 41 } 42 }</pre>
--	---

Sexto Commit.

Showing 2 changed files with 13 additions and 4 deletions.

<p>Filter changed files</p> <p>▼ fase01/lab05</p> <ul style="list-style-type: none"> Soldado.java Videoluego2.java 	<pre> 85 private static Soldado soldadoMayorVida(Soldado[][] soldados) { 86 return null; 87 } 88 89 private static double promedioPuntosVida(Soldado[][] soldados) { 90 return 0; 91 } 92 93 private static int sumaPuntosVida(Soldado[][] soldados) { 94 return 0; 95 }</pre>	<pre> 85 private static Soldado soldadoMayorVida(Soldado[][] soldados) { 86 int idx = 0; 87 for (int i = 1; i < soldados.length; i++) { 88 if (soldados[i].getVida() > soldados[idx].getVida()) 89 idx = i; 90 } 91 return soldados[idx]; 92 } 93 94 private static double promedioPuntosVida(Soldado[][] soldados) { 95 int suma = sumaPuntosVida(soldados); 96 return 1.0 * suma / soldados.length; 97 } 98 99 private static int sumaPuntosVida(Soldado[][] soldados) { 100 int suma = 0; 101 for (int i = 0; i < soldados.length; i++) 102 suma += soldados[i].getVida(); 103 return suma; 104 }</pre>
--	--	---

Séptimo Commit.

Showing 2 changed files with 13 additions and 4 deletions.

- fae01/so05
 - Soldado.java
 - VideoLuego2.java

```

85 private static Soldado soldadoMayorVida(Soldado[] soldados) {
86 -     return null;
87 }
88
89 private static double promedioPuntosVida(Soldado[] soldados) {
90 -     return 0;
91 }
92
93 private static int sumaPuntosVida(Soldado[] soldados) {
94 -     return 0;
95 }

```

```

85 private static Soldado soldadoMayorVida(Soldado[] soldados) {
86 +     int idx = 0;
87 +     for (int i = 1; i < soldados.length; i++) {
88 +         if (soldados[i].getVida() > soldados[idx].getVida())
89 +             idx = i;
90 +     }
91 +     return soldados[idx];
92 }
93
94 private static double promedioPuntosVida(Soldado[] soldados) {
95 +     int suma = sumaPuntosVida(soldados);
96 +     return 1.0 * suma / soldados.length;
97 }
98
99 private static int sumaPuntosVida(Soldado[] soldados) {
100 +     int suma = 0;
101 +     for (int i = 0; i < soldados.length; i++)
102 +         suma += soldados[i].getVida();
103 +     return suma;
104 }

```

Octavo Commit.

Showing 1 changed file with 22 additions and 8 deletions.

```

106 - private static void ordenarSoldadosBurbuja(Soldado[] soldados) {
107 }
108
109 - private static void ordenarSoldadosSelección(Soldado[] soldados) {
110 }
111
112 - private static void imprimirSoldados(Soldado[] soldados) {
113 }

```

```

106 + public static void ordenarSoldadosBurbuja(Soldado[] soldados) {
107 +     for (int i = 0; i < soldados.length - 1; i++) {
108 +         for (int j = 0; j < soldados.length - i - 1; j++) {
109 +             int vida1 = soldados[j].getVida();
110 +             int vida2 = soldados[j + 1].getVida();
111 +             if (vida1 < vida2)
112 +                 intercambiar(soldados, j, j + 1);
113 +         }
114 +     }
115 + }
116 +
117 + public static void ordenarSoldadosSelección(Soldado[] soldados) {
118 }
119
120 + public static void imprimirSoldados(Soldado[] soldados) {
121 }
122
123 + public static void intercambiar(Soldado[] soldados, int i, int j) {
124 +     Soldado t = soldados[i];
125 +     soldados[i] = soldados[j];
126 +     soldados[j] = t;
127 }

```

Noveno Commit.

Showing 1 changed file with 10 additions and 0 deletions.

```

116
117 public static void ordenarSoldadosSelección(Soldado[] soldados) {
118 }

```

```

116 public static void ordenarSoldadosSelección(Soldado[] soldados) {
117 +     for (int i = 0; i < soldados.length - 1; i++) {
118 +         int idx = i;
119 +         for (int j = i + 1; j < soldados.length; j++) {
120 +             int vida1 = soldados[j].getVida();
121 +             int vida2 = soldados[idx].getVida();
122 +             if (vida1 > vida2)
123 +                 idx = j;
124 +         }
125 +         intercambiar(soldados, i, idx);
126 +     }
127 }
128 }

```

Decimo Commit.

5. Código desarrollado

Soldado.java

```
1 package fase01.lab05;
2
3 public class Soldado {
4     private String nombre;
5     private int vida;
6
7     public Soldado(String nombre, int vida) {
8         setNombre(nombre);
9         setVida(vida);
10    }
11
12    public void setNombre(String nombre) {
13        this.nombre = nombre;
14    }
15
16    public void setVida(int vida) {
17        this.vida = vida;
18    }
19
20    public String getNombre() {
21        return nombre;
22    }
23
24    public int getVida() {
25        return vida;
26    }
27
28    public String toString() {
29        return (nombre + " - " + vida);
30    }
31 }
```

- Clase que guarda nombre y vida del soldado.
- Posee tanto setters como getters para todos los atributos.
- Posee el metodo toString() para poder imprimir el objeto.

VideoJuego2.java

```
1 package fase01.lab05;
2
3 import java.util.Random;
4
5 public class VideoJuego2 {
6     public static void main(String[] args) {
7         Random r = new Random();
8         int cantidad = r.nextInt(10) + 1;
9         Soldado[][] tablero = new Soldado[10][10];
10        Soldado[] soldados = new Soldado[cantidad];
11        inicializarSoldados(tablero, soldados);
12        imprimirTablero(tablero);
13        System.out.printf("Soldado con mayor vida: %s%n", soldadoMayorVida(soldados));
14        System.out.printf("Promedio de puntos de vida: %f%n", promedioPuntosVida(soldados));
15        System.out.printf("Nivel de vida del ejercito: %d%n", sumaPuntosVida(soldados));
16        Soldado[] soldados2 = new Soldado[soldados.length];
17        Soldado[] soldados3 = new Soldado[soldados.length];
18        System.arraycopy(soldados, 0, soldados2, 0, soldados.length);
19        System.arraycopy(soldados, 0, soldados3, 0, soldados.length);
20        System.out.println();
21        ordenarSoldadosBurbuja(soldados2);
22        imprimirSoldados(soldados2);
23        System.out.println();
24        ordenarSoldadosSeleccion(soldados3);
25        imprimirSoldados(soldados3);
26    }
27
28    public static void inicializarSoldados(Soldado[][] tablero, Soldado[] soldados) {
29        Random r = new Random();
30        for (int i = 0; i < soldados.length; i++) {
31            String nombre = "Soldado" + i;
32            int vida = r.nextInt(5) + 1;
33            int fila, columna;
34            do {
35                fila = r.nextInt(10);
36                columna = r.nextInt(10);
37            } while (tablero[fila][columna] != null);
38            Soldado soldado = new Soldado(nombre, vida);
39            tablero[fila][columna] = soldado;
40            soldados[i] = soldado;
41        }
42    }
43
44    public static void imprimirTablero(Soldado[][] soldados) {
45        System.out.print(generarEncabezado(soldados));
46        String separacion = generarSeparacion(soldados);
47        for (int i = 0; i < soldados.length; i++) {
48            System.out.print(separacion);
49            System.out.print(generarFila(soldados, i));
50        }
51        System.out.print(separacion);
52    }
53
54    public static String generarEncabezado(Soldado[][] soldados) {
55        String encabezado = "\t";
```



```
56     for (int i = 0; i < soldados.length; i++)
57         encabezado += (" " + intToChar(i + 1) + " ");
58     encabezado += " \n";
59     return encabezado;
60 }
61
62 public static String generarSeparacion(Soldado[] [] soldados) {
63     String fila = "\t";
64     for (int i = 0; i < soldados[0].length; i++)
65         fila += "----";
66     fila += "-\n";
67     return fila;
68 }
69
70 public static String generarFila(Soldado[] [] soldados, int f) {
71     String fila = (f + 1) + "\t";
72     for (int i = 0; i < soldados[f].length; i++) {
73         fila += "| ";
74         Soldado soldado = soldados[f][i];
75         if (soldado != null)
76             fila += soldado.getNombre().charAt(soldado.getNombre().length() - 1);
77         else
78             fila += " ";
79         fila += " ";
80     }
81     fila += "|\n";
82     return fila;
83 }
84
85 public static Soldado soldadoMayorVida(Soldado[] soldados) {
86     int idx = 0;
87     for (int i = 1; i < soldados.length; i++) {
88         if (soldados[i].getVida() > soldados[idx].getVida())
89             idx = i;
90     }
91     return soldados[idx];
92 }
93
94 public static double promedioPuntosVida(Soldado[] soldados) {
95     int suma = sumaPuntosVida(soldados);
96     return 1.0 * suma / soldados.length;
97 }
98
99 public static int sumaPuntosVida(Soldado[] soldados) {
100     int suma = 0;
101     for (int i = 0; i < soldados.length; i++)
102         suma += soldados[i].getVida();
103     return suma;
104 }
105
106 public static void ordenarSoldadosBurbuja(Soldado[] soldados) {
107     for (int i = 0; i < soldados.length - 1; i++) {
108         for (int j = 0; j < soldados.length - i - 1; j++) {
109             int vida1 = soldados[j].getVida();
110             int vida2 = soldados[j + 1].getVida();
111             if (vida1 < vida2)
```

```
112         intercambiar(soldados, j, j + 1);
113     }
114 }
115 }
116
117 public static void ordenarSoldadosSeleccion(Soldado[] soldados) {
118     for (int i = 0; i < soldados.length - 1; i++) {
119         int idx = i;
120         for (int j = i + 1; j < soldados.length; j++) {
121             int vida1 = soldados[j].getVida();
122             int vida2 = soldados[idx].getVida();
123             if (vida1 > vida2)
124                 idx = j;
125         }
126         intercambiar(soldados, i, idx);
127     }
128 }
129
130 public static void imprimirSoldados(Soldado[] soldados) {
131     for (Soldado soldado : soldados)
132         System.out.println(soldado);
133 }
134
135 public static void intercambiar(Soldado[] soldados, int i, int j) {
136     Soldado t = soldados[i];
137     soldados[i] = soldados[j];
138     soldados[j] = t;
139 }
140
141 public static char intToChar(int n) {
142     return (char) (n + 'A' - 1);
143 }
144 }
```

- Método inicializarSoldados() crea a los soldados, los ubica en el tablero y los guarda en un arreglo de soldados por separado.
- Método imprimirTablero() imprime el tablero con ayuda de los métodos auxiliares generarEncabezado(), generarSeparacion() y generarFila(), ubicando a los soldados por su numero.
- Método soldadoMayorVida() retorna el soldado con mayor vida.
- Método promedioPuntosVida() retorna el promedio de los puntos de vida de todos los soldados.
- Método sumaPuntosVida() retorna la suma de los puntos de vida de todos los soldados.
- Se crean dos copias del arreglo de soldados para demostrar los 2 ordenamientos.
- Método ordenarSoldadosBurbuja() ordena los soldados por vida de mayor a menor, usando ordenamiento burbuja.
- Método ordenarSoldadosSeleccion() ordena los soldados por vida de mayor a menor, usando ordenamiento selección.
- Método imprimirSoldados() imprime los soldados del ejercito.

6. Ejecución del código

VideoJuego2.java

```

1      A B C D E F G H I J
2      -----
3  1    | | | | | | | | | |
4      -----
5  2    | | | | | | | | 0 | |
6      -----
7  3    | | | 8 | | | | | | 4 |
8      -----
9  4    | | | | | | | | | |
10     -----
11  5    | | | | | | | | | |
12     -----
13  6    | | | | | 1 | 5 | | | |
14     -----
15  7    | | | | | | | | 2 | |
16     -----
17  8    | | | | | 3 | | | | |
18     -----
19  9    | | | | | | | | | |
20     -----
21 10    | | | | | 6 | 7 | | | |
22     -----
23 Soldado con mayor vida: Soldado2 - 5
24 Promedio de puntos de vida: 3.333333
25 Nivel de vida del ejercito: 30
26
27 Soldado2 - 5
28 Soldado8 - 5
29 Soldado0 - 4
30 Soldado3 - 4
31 Soldado4 - 3
32 Soldado6 - 3
33 Soldado7 - 3
34 Soldado1 - 2
35 Soldado5 - 1
36
37 Soldado2 - 5
38 Soldado8 - 5
39 Soldado0 - 4
40 Soldado3 - 4
41 Soldado4 - 3
42 Soldado6 - 3
43 Soldado7 - 3
44 Soldado1 - 2
45 Soldado5 - 1

```

7. Estructura de laboratorio 05

- El contenido que se entrega en este laboratorio es el siguiente:

```
lab05/  
|--- Soldado.java  
|--- VideoJuego2.java  
|--- ejec01.bash  
|--- Informe.tex  
|--- Informe.pdf  
|--- img  
    |--- logo_abet.png  
    |--- logo_episunsa.png  
    |--- logo_unsa.jpg  
    |--- commit01.jpg  
    |--- commit02.jpg  
    |--- commit03.jpg  
    |--- commit04.jpg  
    |--- commit05.jpg  
    |--- commit06.jpg  
    |--- commit07.jpg  
    |--- commit08.jpg  
    |--- commit09.jpg  
    |--- commit10.jpg
```

8. Rúbricas

8.1. Entregable Informe

Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y fácil de leer.

8.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobatoria, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Niveles de desempeño

Puntos	Nivel			
	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

Rúbrica para contenido del Informe y demostración

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	4	
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	1.5	
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 estan dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	1.5	
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		18	

9. Referencias

- Aedo, M. y Castro, E. (2021). FUNDAMENTOS DE PROGRAMACIÓN 2 - Tópicos de Programación Orientada a Objetos. Editorial UNSA.