


	<b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b> <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b> <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b>	
<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> 1

## INFORME DE LABORATORIO

### (formato estudiante)

INFORMACIÓN BÁSICA					
<b>ASIGNATURA:</b>	<i>Fundamentos de la programación 2</i>				
<b>TÍTULO DE LA PRÁCTICA:</b>	<i>Combinando Arreglos Estándar y ArrayList</i>				
<b>NÚMERO DE PRÁCTICA:</b>	<i>07</i>	<b>AÑO LECTIVO:</b>	<i>2023</i>	<b>NRO. SEMESTRE:</b>	<i>2do Semestre</i>
<b>FECHA DE PRESENTACIÓN</b>	<i>25/12/2023</i>	<b>HORA DE PRESENTACIÓN</b>	<i>hh/mm/ss</i>		
<b>INTEGRANTE (s)</b> <i>Juan Diego Gutiérrez Ccama</i>				<b>NOTA (0-20)</b>	<i>Nota colocada por el docente</i>
<b>DOCENTE(s):</b> <i>Linno Jose Pinto Oppe</i>					

RESULTADOS Y PRUEBAS
<b>I. EJERCICIOS RESUELTOS:</b>

```
Prueba.java  Database.java  *Soldado.java x  *VideoJuego3.java
1  //Clase soldado
2  public class Soldado {
3      private String nombre;
4      private int puntosVida;
5      private int fila;
6      private int columna;
7
8      public Soldado(String nombre, int puntosVida, int fila, int columna) {
9          this.nombre = nombre;
10         this.puntosVida = puntosVida;
11         this.fila = fila;
12         this.columna = columna;
13     }
14
15     public String getNombre() {
16         return nombre;
17     }
18
19     public int getPuntosVida() {
20         return puntosVida;
21     }
22
23     public int getFila() {
24         return fila;
25     }
26
27     public int getColumna() {
28         return columna;
29     }
30
31     public static String generarNombre(int indice, int ejercito) {
32         return "Soldado" + ejercito + "X" + indice;
33     }
34
35     public static int generarPuntosVida() {
36         int gv = (int) (Math.random()*5 + 1);
37         return gv;
38     }
39     public String toString() {
40         return nombre + "(" + puntosVida + ")";
41     }
42 }
43
```

## II. PRUEBAS

```
Prueba.java × Database.java *Soldado.java *VideoJuego3.java Soldado.java VideoJuego4.java ×
1 public class VideoJuego4 {
2     public static void main(String[] args) {
3         int filas = 10;
4         int columnas = 10;
5         Soldado[][] tablero = new Soldado[filas][columnas];
6
7         Soldado[] soldados01 = new Soldado[genNumSol()];
8         Soldado[] soldados02 = new Soldado[genNumSol()];
9
10        for (int i = 0; i < filas; i++) {
11            for (int j = 0; j < columnas; j++) {
12                int ejercito = i % 2;
13                Soldado soldado = crearSoldado(i * columnas + j, ejercito);
14                tablero[i][j] = soldado;
15            }
16        }
17
18        imprimirTablero(tablero);
19
20        for (int i = 0; i < filas; i++) {
21            Soldado[] ejercito = tablero[i];
22            ordenarPorBurbuja(ejercito);
23            ordenarPorQuicksort(ejercito, 0, ejercito.length - 1);
24        }
25
26        int sumaTotalEjercito0 = calcularSumaTotalPuntosVida(tablero[0]);
27        int sumaTotalEjercito1 = calcularSumaTotalPuntosVida(tablero[1]);
28
29        if (sumaTotalEjercito0 > sumaTotalEjercito1) {
30            System.out.println("¡El Ejército 0 gana la batalla!");
31        } else if (sumaTotalEjercito1 > sumaTotalEjercito0) {
32            System.out.println("¡El Ejército 1 gana la batalla!");
33        } else {
34            System.out.println("¡La batalla termina en empate!");
35        }
36    }
37
38    private static void imprimirTablero(Soldado[][] tablero) {
39        for (int i = 0; i < tablero.length; i++) {
40            for (int j = 0; j < tablero[i].length; j++) {
41                imprimirCasilla(tablero[i][j]);
42            }
43            System.out.println(); // Agregar salto de línea después de cada fila
```

```
Prueba.java | Database.java | *Soldado.java | *VideoJuego3.java | Soldado.java | VideoJuego4.java X
44     }
45 }
46
47 private static void imprimirCasilla(Soldado soldado) {
48     System.out.print("| " + soldado.getNombre() + " ");
49 }
50
51 private static void ordenarPorBurbuja(Soldado[] soldados) {
52     int n = soldados.length;
53     for (int i = 0; i < n - 1; i++) {
54         for (int j = 0; j < n - i - 1; j++) {
55             if (soldados[j].getPuntosVida() < soldados[j + 1].getPuntosVida()) {
56                 Soldado temp = soldados[j];
57                 soldados[j] = soldados[j + 1];
58                 soldados[j + 1] = temp;
59             }
60         }
61     }
62 }
63
64 private static void ordenarPorQuicksort(Soldado[] soldados, int inicio, int fin) {
65     if (inicio < fin) {
66         int indicePivote = particionQuicksort(soldados, inicio, fin);
67         ordenarPorQuicksort(soldados, inicio, indicePivote - 1);
68         ordenarPorQuicksort(soldados, indicePivote + 1, fin);
69     }
70 }
71
72 private static int particionQuicksort(Soldado[] soldados, int inicio, int fin) {
73     Soldado pivote = soldados[fin];
74     int i = inicio - 1;
75
76     for (int j = inicio; j < fin; j++) {
77         if (soldados[j].getPuntosVida() >= pivote.getPuntosVida()) {
78             i++;
79             Soldado temp = soldados[i];
80             soldados[i] = soldados[j];
81             soldados[j] = temp;
82         }
83     }
84
85     Soldado temp = soldados[i + 1];
86     soldados[i + 1] = soldados[fin];
```

```

86         soldados[i + 1] = soldados[fin];
87         soldados[fin] = temp;
88
89         return i + 1;
90     }
91
92     private static int calcularSumaTotalPuntosVida(Soldado[] soldados) {
93         int suma = 0;
94         for (Soldado soldado : soldados) {
95             suma += soldado.getPuntosVida();
96         }
97         return suma;
98     }
99
100    private static Soldado crearSoldado(int indice, int ejercito) {
101        String nombre = Soldado.generarNombre(indice, ejercito);
102        int puntosVida = Soldado.generarPuntosVida();
103        int fila = indice / 10;
104        int columna = indice % 10;
105        return new Soldado(nombre, puntosVida, fila, columna);
106    }
107
108    public static int genNumSol() {
109        return (int) (Math.random() * 10 + 1);
110    }
111 }

```

#### IV. CUESTIONARIO:

```

<terminated> VideoJuego4 [Java Application] C:\Users\UJKMjuandi\Downloads\eclipse-java-2023-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6
Soldado0X0 | Soldado0X1 | Soldado0X2 | Soldado0X3 | Soldado0X4 | Soldado0X5 | Soldado0X6 | Soldado0X7 | Soldado0X8 | Soldado0X9
Soldado1X10 | Soldado1X11 | Soldado1X12 | Soldado1X13 | Soldado1X14 | Soldado1X15 | Soldado1X16 | Soldado1X17 | Soldado1X18 | Soldado1X19
Soldado0X20 | Soldado0X21 | Soldado0X22 | Soldado0X23 | Soldado0X24 | Soldado0X25 | Soldado0X26 | Soldado0X27 | Soldado0X28 | Soldado0X29
Soldado1X30 | Soldado1X31 | Soldado1X32 | Soldado1X33 | Soldado1X34 | Soldado1X35 | Soldado1X36 | Soldado1X37 | Soldado1X38 | Soldado1X39
Soldado0X40 | Soldado0X41 | Soldado0X42 | Soldado0X43 | Soldado0X44 | Soldado0X45 | Soldado0X46 | Soldado0X47 | Soldado0X48 | Soldado0X49
Soldado1X50 | Soldado1X51 | Soldado1X52 | Soldado1X53 | Soldado1X54 | Soldado1X55 | Soldado1X56 | Soldado1X57 | Soldado1X58 | Soldado1X59
Soldado0X60 | Soldado0X61 | Soldado0X62 | Soldado0X63 | Soldado0X64 | Soldado0X65 | Soldado0X66 | Soldado0X67 | Soldado0X68 | Soldado0X69
Soldado1X70 | Soldado1X71 | Soldado1X72 | Soldado1X73 | Soldado1X74 | Soldado1X75 | Soldado1X76 | Soldado1X77 | Soldado1X78 | Soldado1X79
Soldado0X80 | Soldado0X81 | Soldado0X82 | Soldado0X83 | Soldado0X84 | Soldado0X85 | Soldado0X86 | Soldado0X87 | Soldado0X88 | Soldado0X89
Soldado1X90 | Soldado1X91 | Soldado1X92 | Soldado1X93 | Soldado1X94 | Soldado1X95 | Soldado1X96 | Soldado1X97 | Soldado1X98 | Soldado1X99
El Ejército 1 gana la batalla!

```

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

	<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> 6

## CONCLUSIONES

Aprecio la oportunidad de destacar que he desarrollado habilidades significativas al trabajar con estructuras de datos tanto mediante el uso de arrays estándar como de ArrayList en el contexto de mi programación. Este logro refleja mi capacidad para no solo manipular datos de manera eficiente sino también para realizar tareas que abarcan desde las más fundamentales hasta las más complejas.

La combinación de arrays estándar y ArrayList ha ampliado de manera notable mi flexibilidad y eficacia en la gestión de datos estructurados. Esta estrategia me ha permitido organizar y estructurar información de manera más efectiva, proporcionando una base robusta para la ejecución de operaciones avanzadas en el ámbito de la programación.

Es crucial destacar que esta capacidad va más allá de la simple manipulación de datos. La integración de arrays estándar y ArrayList ha capacitado no solo para acciones básicas, como la agregación y eliminación de elementos, sino también para llevar a cabo funciones más avanzadas que implican un análisis detallado y una manipulación sofisticada de datos. La combinación de estas dos estructuras de datos me ha permitido abordar con éxito problemas más complejos y satisfacer requisitos más exigentes en el desarrollo de software.

## METODOLOGÍA DE TRABAJO



- 1.- *Leer los enunciados cuidadosamente para saber que requiere el problemas.*
- 2.- *Empezar a hacer las clase soldado*
- 3.- *Realizar el Videojuego4*

## REFERENCIAS Y BIBLIOGRAFÍA

[1] M. A. Lopez, E. Castro Gutierrez, *Fundamentos de la Programación 2 Topicos de Programación Orientada a Objetos*. Arequipa: UNAS, 2021

## RUBRICA PARA EL CONTENIDO DEL INFORME Y DEMOSTRACIÓN

El alumno debe marcar o dejar en blanco en celdas de la columna Checklist si cumplió con el ítem correspondiente.

	<b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b> <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b> <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b>	
<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> 7

Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.



El alumno debe autocalificarse en la columna Estudiante de acuerdo a la siguiente tabla:

Tabla 1: Niveles de desempeño

Nivel				
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	3	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1	
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	0	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	3	
TOTAL		20		14	

Tabla 2: Rúbrica para contenido del Informe y demostración

 <b>INGENIERIA</b> <b>ISTEMAS</b>	<b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b> <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</b> <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b>	 <b>ABET</b>
<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> 8