


	<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>HashMap</i>				
<b>NÚMERO DE PRÁCTICA:</b>	<i>08</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>09/20/00</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

```
Soldado.java VideoJuego5.java Soldado.java
1 import java.util.HashMap;
2 import java.util.Map;
3
4 public class VideoJuego5 {
5     public static void main(String[] args) {
6         int totalSoldados = 100; // 10 filas x 10 columnas
7         Map<Integer, Soldado> soldadosMap = new HashMap<>();
8
9         for (int i = 0; i < totalSoldados; i++) {
10             int ejercito = i % 2;
11             Soldado soldado = crearSoldado(i, ejercito);
12             soldadosMap.put(i, soldado);
13         }
14
15         imprimirTablero(soldadosMap);
16
17         int sumaTotalEjercito0 = calcularSumaTotalPuntosVida(soldadosMap, 0, totalSoldados / 2);
18         int sumaTotalEjercito1 = calcularSumaTotalPuntosVida(soldadosMap, totalSoldados / 2, totalSoldados);
19
20         determinarGanador(sumaTotalEjercito0, sumaTotalEjercito1);
21     }
22
23     private static void imprimirTablero(Map<Integer, Soldado> soldadosMap) {
24         int columnas = 10;
25         for (int i = 0; i < soldadosMap.size(); i++) {
26             imprimirCasilla(soldadosMap.get(i));
27             if ((i + 1) % columnas == 0) {
28                 System.out.println(); // Salto de línea después de cada fila
29             }
30         }
31     }
32
33     private static void imprimirCasilla(Soldado soldado) {
34         System.out.print("| " + soldado.getNombre() + " ");
35     }
36
37     private static int calcularSumaTotalPuntosVida(Map<Integer, Soldado> soldadosMap, int inicio, int fin) {
38         int suma = 0;
39         for (int i = inicio; i < fin; i++) {
40             suma += soldadosMap.get(i).getPuntosVida();
41         }
42         return suma;
43     }
44 }
```

**III.**

```

43     }
44
45     private static void determinarGanador(int sumaTotalEjercito0, int sumaTotalEjercito1) {
46         if (sumaTotalEjercito0 > sumaTotalEjercito1) {
47             System.out.println("¡El Ejército 0 gana la batalla!");
48         } else if (sumaTotalEjercito1 > sumaTotalEjercito0) {
49             System.out.println("¡El Ejército 1 gana la batalla!");
50         } else {
51             System.out.println("¡La batalla termina en empate!");
52         }
53     }
54
55     private static Soldado crearSoldado(int indice, int ejercito) {
56         String nombre = Soldado.generarNombre(indice, ejercito);
57         int puntosVida = Soldado.generarPuntosVida();
58         return new Soldado(nombre, puntosVida, 0, 0);
59     }
60 }

```

#### IV. CUESTIONARIO:

```


<terminated> VideoJuego4 [Java Application] C:\Users\UJKMjuandi\Downloads\eclipse-java-ZU23-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.b
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>

### CONCLUSIONES

Aprecio la oportunidad de destacar que he desarrollado habilidades significativas al trabajar con diversas estructuras de datos, incluyendo arrays estándar y HashMap en el contexto de mi programación. Este logro refleja mi capacidad para no solo manipular datos de manera eficiente, sino también para llevar a cabo tareas que abarcan desde las más fundamentales hasta las más complejas.

	<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> 5

La incorporación del HashMap ha ampliado notablemente 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 del HashMap me ha capacitado no solo para realizar 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 incorporación de esta estructura de datos específica 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 Videojuego5*

### 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.

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

	<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> 6</p>

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	2	
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		13	

Tabla 2: Rúbrica para contenido del Informe y demostración