

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 1</p>

## INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	Fundamentos de la Programación 2				
TÍTULO DE LA PRÁCTICA:	Practica de Laboratorio 3: Arreglos de Objetos				
NÚMERO DE PRÁCTICA:	3	AÑO LECTIVO:	2024	NRO. SEMESTRE:	Segundo
FECHA DE PRESENTACIÓN	05/10/2024	HORA DE PRESENTACIÓN	23:59		
INTEGRANTE (s) Santiago Alonso Quintanilla Chávez				NOTA (0-20)	
DOCENTE(s): Ing. Lino Jose Pinto Oppe					

RESULTADOS Y PRUEBAS
<p><b>I. EJERCICIOS RESUELTOS:</b></p> <p>Cree un Proyecto llamado Laboratorio3</p> <p>Usted deberá agregar las clases Nave.java y DemoBatalla.java</p> <p>1. Analice, complete y pruebe el Código de la clase DemoBatalla</p>

```
1 public class Nave {
2     private String nombre;
3     private int fila;
4     private String columna;
5     private boolean estado;
6     private int puntos;
7
8     public void setNombre(String n){
9         nombre=n;
10    }
11    public void setFila(int f){
12        fila=f;
13    }
14    public void setColumna(String c){
15        columna=c;
16    }
17    public void setEstado(boolean e){
18        estado=e;
19    }
20    public void setPuntos(int p){
21        puntos=p;
22    }
23    public String getNombre(){
24        return nombre;
25    }
26    public int getFila(){
27        return fila;
28    }
29    public String getColumna(){
30        return columna;
31    }
32    public boolean getEstado(){
33        return estado;
34    }
35    public int getPuntos(){
36        return puntos;
37    }
38    public String toString() {
39        return "-Nombre: "+nombre+"\t-Posicion: (" +fila+", "+columna+")\t-Estado: "+estado+"\t-Puntos: "+puntos;
40    }
41 }
```

```
1 import java.util.*;
2 public class DemoBatalla {
3     public static void main(String[] args){
4         Nave [] misNaves=new Nave[5];
5         Scanner sc=new Scanner(System.in);
6         String nomb, col;
7         int fil, punt;
8         boolean est;
9         for (int i=0; i<misNaves.length;i++) {
10             System.out.println("Nave "+(i+1));
11             System.out.print("Nombre: ");
12             nomb=sc.next();
13             System.out.print("Fila: ");
14             fil=sc.nextInt();
15             System.out.print("Columna: ");
16             col=sc.next();
17             System.out.print("Estado: ");
18             est=sc.nextBoolean();
19             System.out.print("Puntos: ");
20             punt=sc.nextInt();
21
22             misNaves[i]=new Nave();
23
24             misNaves[i].setNombre(nomb);
25             misNaves[i].setFila(fil);
26             misNaves[i].setColumna(col);
27             misNaves[i].setEstado(est);
28             misNaves[i].setPuntos(punt);
29         }
30         System.out.println("\nNaves creadas: ");
31         mostrarNaves(misNaves);
32         mostrarPorNombre(misNaves);
33         mostrarPorPuntos(misNaves);
34         System.out.println("\nNave con mayor número de puntos: \n-->"+mostrarMayorPuntos(misNaves));
35         imprimir(metodoAleatorio(misNaves));
36     }
37     public static void mostrarNaves(Nave[] flota){
38         System.out.println("Naves de la flota: ");
39         for (int i=0;i<flota.length;i++){
40             System.out.println("-->"+flota[i]);
41         }
42     }
```

```
43 public static void mostrarPorNombre(Nave[] flota){
44     Scanner scan=new Scanner(System.in);
45     System.out.print("Nombre de la nave: ");
46     String nomb=scan.next();
47     Boolean verif=true;
48     for (int i=0;i<flota.length;i++){
49         if (nomb.equals(flota[i].getNombre())){
50             System.out.println("-->" + flota[i]);
51             verif=false;
52         }
53     }
54     if (verif){
55         System.out.println("No se encontro nave");
56     }
57 }
58 public static void mostrarPorPuntos(Nave[] flota){
59     Scanner scan=new Scanner(System.in);
60     System.out.print("Ingrese límite de puntos: ");
61     int puntos=scan.nextInt();
62     Boolean verif=true;
63     for (int i=0;i<flota.length;i++){
64         int comparacion=flota[i].getPuntos();
65         if (comparacion<=puntos){
66             System.out.println("-->" + flota[i]);
67             verif=false;
68         }
69     }
70     if (verif){
71         System.out.println("No se encontro nave");
72     }
73 }
74 public static Nave mostrarMayorPuntos(Nave[] flota){
75     int mayor=0, id=0;
76     for (int i=0;i<flota.length;i++){
77         int comparacion=flota[i].getPuntos();
78         if (comparacion>mayor){
79             mayor=comparacion;
80             id=i;
81         }
82     }
83     Nave naveMayor=flota[id];
84     return naveMayor;
85 }
```

```
86 public static Nave[] metodoAleatorio(Nave[] flota){
87     Random rand=new Random();
88     Nave [] Aleatorio=new Nave[flota.length];
89     ArrayList<Integer> indices=new ArrayList<Integer>();
90     for (int i=0;i<flota.length;i++) {
91         int indice=rand.nextInt(flota.length);
92         while (indices.contains(indice)) {
93             indice=rand.nextInt(flota.length);
94         }
95         indices.add(indice);
96         Aleatorio[i]=flota[indice];
97     }
98     return Aleatorio;
99 }
100 public static void imprimir(Nave[] flota) {
101     for (int i=0;i<flota.length;i++) {
102         System.out.println(flota[i]);
103     }
104 }
105 }
```

```
Nave 1
Nombre: 1raNave
Fila: 3
Columna: 6
Estado: true
Puntos: 45
Nave 2
Nombre: 2daNave
Fila: 4
Columna: 5
Estado: true
Puntos: 23
Nave 3
Nombre: 3raNave
Fila: 1
Columna: 7
Estado: false
Puntos: 57
Nave 4
Nombre: 4taNave
Fila: 6
Columna: 2
Estado: true
Puntos: 21
Nave 5
Nombre: 5taNave
Fila: 9
Columna: 4
Estado: true
Puntos: 39
```

```
Naves creadas:
Naves de la flota:
-->-Nombre: 1raNave      -Posicion: (3,6)      -Estado: true  -Puntos: 45
-->-Nombre: 2daNave      -Posicion: (4,5)      -Estado: true  -Puntos: 23
-->-Nombre: 3raNave      -Posicion: (1,7)      -Estado: false -Puntos: 57
-->-Nombre: 4taNave      -Posicion: (6,2)      -Estado: true  -Puntos: 21
-->-Nombre: 5taNave      -Posicion: (9,4)      -Estado: true  -Puntos: 39
Nombre de la nave: 3raNave
-->-Nombre: 3raNave      -Posicion: (1,7)      -Estado: false -Puntos: 57
Ingrese límite de puntos: 40
|-->-Nombre: 2daNave      -Posicion: (4,5)      -Estado: true  -Puntos: 23
-->-Nombre: 4taNave      -Posicion: (6,2)      -Estado: true  -Puntos: 21
-->-Nombre: 5taNave      -Posicion: (9,4)      -Estado: true  -Puntos: 39

Nave con mayor número de puntos:
-->-Nombre: 3raNave      -Posicion: (1,7)      -Estado: false -Puntos: 57
-Nombre: 5taNave        -Posicion: (9,4)      -Estado: true  -Puntos: 39
-Nombre: 2daNave        -Posicion: (4,5)      -Estado: true  -Puntos: 23
-Nombre: 3raNave        -Posicion: (1,7)      -Estado: false -Puntos: 57
-Nombre: 1raNave        -Posicion: (3,6)      -Estado: true  -Puntos: 45
-Nombre: 4taNave        -Posicion: (6,2)      -Estado: true  -Puntos: 21
```

2. Solucionar la Actividad 4 de la Práctica 1 pero usando arreglo de objetos

```
1 public class Soldado {
2     private String nombre;
3     private int nivelVida;
4     private int orden;
5
6     public void setNombre(String n){
7         nombre=n;
8     }
9     public void setnivelVida(int v) {
10         nivelVida=v;
11     }
12     public void setOrden(int o) {
13         orden=o;
14     }
15     public String getNombre(){
16         return nombre;
17     }
18     public int getnivelVida() {
19         return nivelVida;
20     }
21     public int getOrden() {
22         return orden;
23     }
24     public String toString() {
25         return "Soldado "+orden+": \n-Nombre: "+nombre+"\t-Nivel de Vida: "+nivelVida;
26     }
27 }
```

```
1 import java.util.*;
2 public class Laboratorio3_Ejercicio2 {
3     public static void main(String[] args) {
4         Scanner scan=new Scanner(System.in);
5         Random rand=new Random();
6         Soldado soldados[]=new Soldado[5];
7         for (int i=0;i<5;i++) {
8             soldados[i]=new Soldado();
9             System.out.print("Ingrese el nombre del soldado "+(i+1)+" : ");
10             String nombre=scan.next();
11
12             soldados[i].setNombre(nombre);
13             soldados[i].setnivelVida(rand.nextInt(5)+1);
14             soldados[i].setOrden(i+1);
15         }
16         System.out.println("=====");
17         System.out.println("Datos de los soldados:");
18         for (int k=0;k<5;k++) {
19             System.out.println(soldados[k]);
20         }
21     }
22 }
```

	<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>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 7

```

Ingrese el nombre del soldado 1 : Fernando
Ingrese el nombre del soldado 2 : Maximo
Ingrese el nombre del soldado 3 : Felipe
Ingrese el nombre del soldado 4 : Armando
Ingrese el nombre del soldado 5 : Miguel
=====
Datos de los soldados:
Soldado 1:
-Nombre: Fernando      -Nivel de Vida: 4
Soldado 2:
-Nombre: Maximo -Nivel de Vida: 2
Soldado 3:
-Nombre: Felipe -Nivel de Vida: 5
Soldado 4:
-Nombre: Armando      -Nivel de Vida: 4
Soldado 5:
-Nombre: Miguel -Nivel de Vida: 3

```

3. Solucionar la Actividad 5 de la Práctica 1 pero usando arreglos de objetos

```

1 public class Soldado {
2     private String nombre;
3     private int nivelVida;
4     private int orden;
5
6     public void setNombre(String n){
7         nombre=n;
8     }
9     public void setnivelVida(int v) {
10        nivelVida=v;
11    }
12    public void setOrden(int o) {
13        orden=o;
14    }
15    public String getNombre(){
16        return nombre;
17    }
18    public int getnivelVida() {
19        return nivelVida;
20    }
21    public int getOrden() {
22        return orden;
23    }
24    public String toString() {
25        return "Soldado "+orden+": \n-Nombre: "+nombre+"\t-Nivel de Vida: "+nivelVida;
26    }
27 }

```

```
1 import java.util.*;
2 public class Laboratorio3_Ejercicio3 {
3     public static void main(String[] args) {
4         Random rand=new Random();
5         int n=rand.nextInt(5)+1;
6         int m=rand.nextInt(5)+1;
7         Soldado ejercito01[]=inicializar(n);
8         Soldado ejercito02[]=inicializar(m);
9         System.out.println("Datos del ejercito 1: ");
10        imprimir(ejercito01);
11        System.out.println("Datos del ejercito 2: ");
12        imprimir(ejercito02);
13        System.out.println("=====");
14        System.out.println("El ejercito ganador fue: ");
15        if (n>m) {
16            System.out.println("Ejercito 1!\nDatos del ejercito:");
17            imprimir(ejercito01);
18        }
19        if (m>n) {
20            System.out.println("Ejercito 2!\nDatos del ejercito:");
21            imprimir(ejercito02);
22        }
23        if (m==n) {
24            System.out.println("Hubo empate!\nDatos del ejercito 1: ");
25            imprimir(ejercito01);
26            System.out.println("Datos del ejercito 2: ");
27            imprimir(ejercito02);
28        }
29    }
30    public static void imprimir(Soldado []arreglo) {
31        for (int k=0;k<arreglo.length;k++) {
32            System.out.println(arreglo[k]);
33        }
34    }

35    public static Soldado[] inicializar(int x) {
36        Random rand=new Random();
37        Soldado ejercito[]=new Soldado[x];
38        for (int i=0;i<x;i++) {
39            ejercito[i]=new Soldado();
40            ejercito[i].setNombre("soldado"+i);
41            ejercito[i].setnivelVida(rand.nextInt(5)+1);
42            ejercito[i].setOrden(i+1);
43        }
44        return ejercito;
45    }
46 }
```



```
Datos del ejercito 1:
Soldado 1:
-Nombre: soldado0      -Nivel de Vida: 2
Datos del ejercito 2:
Soldado 1:
-Nombre: soldado0      -Nivel de Vida: 1
Soldado 2:
-Nombre: soldado1      -Nivel de Vida: 4
Soldado 3:
-Nombre: soldado2      -Nivel de Vida: 3
=====
El ejercito ganador fue:
Ejercito 2!
Datos del ejercito:
Soldado 1:
-Nombre: soldado0      -Nivel de Vida: 1
Soldado 2:
-Nombre: soldado1      -Nivel de Vida: 4
Soldado 3:
-Nombre: soldado2      -Nivel de Vida: 3
```

	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. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4			
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2			
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2			
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2		X	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2			
7. Ortografía	El documento no muestra errores ortográficos.	2		X	
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			
TOTAL		20			