

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

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	<i>Fundamentos de la Programación 2</i>				
TÍTULO DE LA PRÁCTICA:	<i>Practica de Laboratorio 6: ArrayList</i>				
NÚMERO DE PRÁCTICA:	<i>6</i>	AÑO LECTIVO:	<i>2024</i>	NRO. SEMESTRE:	<i>Segundo</i>
FECHA DE PRESENTACIÓN	<i>25/10/2024</i>	HORA DE PRESENTACIÓN	<i>18:20</i>		
INTEGRANTE (s) <i>Santiago Alonso Quintanilla Chávez</i>				NOTA (0-20)	
DOCENTE(s): <i>Ing. Lino Jose Pinto Oppe</i>					

RESULTADOS Y PRUEBAS
<p>I. EJERCICIOS RESUELTOS:</p> <ol style="list-style-type: none"> 1. Cree un Proyecto llamado Laboratorio6 2. Usted deberá crear las dos clases Soldado.java y VideoJuego3.java. Puede reutilizar lo desarrollado en Laboratorios anteriores. 3. Del Soldado nos importa el nombre, puntos de vida, fila y columna (posición en el tablero). 4. El juego se desarrollará en el mismo tablero de los laboratorios anteriores. Pero ahora el tablero debe ser un ArrayList bidimensional. 5. Tendrá 2 Ejércitos. Inicializar el tablero con n soldados aleatorios entre 1 y 10 para cada Ejército. Cada soldado tendrá un nombre autogenerado: Soldado0X1, Soldado1X1, 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 (distinguir los de un ejército de los del otro ejército). Además de los datos del Soldado con mayor vida de cada ejército, el promedio de puntos de vida de todos los soldados creados por ejército, los datos de todos los soldados por ejército en el orden que fueron creados y un ranking de poder de todos los soldados creados por ejército (del que tiene más nivel de vida al que tiene menos) usando 2 diferentes Marco Aedo López 2 algoritmos de ordenamiento. Finalmente, que muestre qué ejército ganará la batalla (indicar la métrica usada para decidir al ganador de la batalla). <ol style="list-style-type: none"> a. Código de la clase "Soldado":

```
1 public class Soldado {
2     private String nombre;
3     private int nivelVida;
4     private int fila;
5     private int columna;
6     private boolean ocupado;
7     private int numEjer;
8     public Soldado(){
9     }
10    public void setNombre(String n){
11        nombre=n;
12    }
13    public void setNivelVida(int v) {
14        nivelVida=v;
15    }
16    public void setFila(int f) {
17        fila=f;
18    }
19    public void setColumna (int c) {
20        columna=c;
21    }
22    public void setOcupado (boolean o) {
23        ocupado=o;
24    }
25    public void setNumEjercito (int e){
26        numEjer=e;
27    }
28    public String getNombre(){
29        return nombre;
30    }
31    public int getNivelVida() {
32        return nivelVida;
33    }
34    public int getFila() {
35        return fila;
36    }
37    public int getColumna() {
38        return columna;
39    }
40    public boolean getOcupado() {
41        return ocupado;
42    }
43    public int getNumEjercito(){
44        return numEjer;
45    }
46    public String toString() {
47        return "-Nombre: "+nombre+"\t-Posición: (" +fila+", "+columna+")\t-Nivel de Vida: "+nivelVida;
48    }
49 }
```

b. Código de clase "Videojuego3":

```

1  import java.util.*;
2  public class Videojuego3 {
3      Run | Debug | Run main | Debug main
4      public static void main(String[] args) {
5          ArrayList<ArrayList<Soldado>> tablero=inicializarTablero();
6          ArrayList<Soldado> ejercito1=new ArrayList<Soldado>();
7          ArrayList<Soldado> ejercito2=new ArrayList<Soldado>();
8
9          inicializarEjercito(tablero, x:1, ejercito1);
10         inicializarEjercito(tablero, x:2, ejercito2);
11         imprimirTablero(tablero);
12
13         Soldado soldMayor1=soldadoMayorVida(ejercito1);
14         Soldado soldMayor2=soldadoMayorVida(ejercito2);
15         System.out.println(x:"Los soldados con mayor vida son:");
16         System.out.println("\n-Ejercito 1: "+soldMayor1+"\n-Ejercito 2: "+soldMayor2);
17         System.out.println(x:"\n_____");
18         Double vida1=promedioNivelVida(ejercito1);
19         Double vida2=promedioNivelVida(ejercito2);
20         System.out.println("El promedio del nivel de vida del ejercito 1 es: "+vida1);
21         System.out.println("El promedio del nivel de vida del ejercito 2 es: "+vida2);
22         imprimirOrdenCreacion(ejercito1);
23         imprimirOrdenCreacion(ejercito2);
24         imprimirRankingVida(ejercito1);
25         imprimirRankingVida(ejercito2);
26         decidirEjercitoGanador(ejercito1, ejercito2);
27     }
28     //Metodo que imprime el tablero creado, según el formato indicado
29     public static void imprimirTablero(ArrayList<ArrayList<Soldado>> arreglo) {
30         System.out.println(x:"\n_____");
31         System.out.println(x:"Tablero de posiciones: ");
32         for (int k=0;k<arreglo.size();k++) {
33             System.out.println(x:"");
34             for (int j=0;j<arreglo.get(k).size();j++) {
35                 boolean ocupado=arreglo.get(k).get(j).getOcupado();
36                 if (ocupado) {
37                     int num=arreglo.get(k).get(j).getNumEjercito();
38                     System.out.print("|_"+num+"_|");
39                 } else {
40                     System.out.print(s:"|_|");
41                 }
42             }
43         }
44         //Metodo que crea un tablero vacio bidimensional de 10x10
45         public static ArrayList<ArrayList<Soldado>> inicializarTablero() {
46             ArrayList<ArrayList<Soldado>> tablero=new ArrayList<ArrayList<Soldado>>();
47             for (int i=0;i<10;i++) {
48                 ArrayList<Soldado> fila=new ArrayList<Soldado>();

```

```
49     for (int j=0;j<10;j++) {
50         Soldado s=new Soldado();
51         s.setOcupado(o:false);
52         fila.add(s);
53     }
54     tablero.add(fila);
55 }
56 return tablero;
57 }
58 //Metodo que crea un ejercito, lo ubica en el mapa (verificando que no se repitan dos soldados en un
59 //mismo lugar), y agragandolos a un ArrayList "ejercito"
60 public static void inicializarEjercito(ArrayList<ArrayList<Soldado>> array, int x, ArrayList<Soldado> ejer){
61     Random rand=new Random();
62     int NumSold=rand.nextInt(bound:10)+1;
63     for (int k=0;k<NumSold;k++) {
64         int fila=rand.nextInt(bound:10);
65         int columna=rand.nextInt(bound:10);
66         boolean ocupado=array.get(fila).get(columna).getOcupado();
67         while (ocupado) {
68             fila=rand.nextInt(bound:10);
69             columna=rand.nextInt(bound:10);
70             ocupado=array.get(fila).get(columna).getOcupado();
71         }
72         String nombre="Soldado"+(k)+"x"+(x);
73         array.get(fila).get(columna).setNombre(nombre);
74         array.get(fila).get(columna).setFila(fila+1);
75         array.get(fila).get(columna).setColumna(columna+1);
76         array.get(fila).get(columna).setNivelVida(rand.nextInt(bound:5)+1);
77         array.get(fila).get(columna).setOcupado(o:true);
78         array.get(fila).get(columna).setNumEjercito(x);
79         ejer.add(array.get(fila).get(columna));
80     }
81 }
82 //Metodo que compara los soldados en un ArrayList "Ejercito" y devuelve el soldado con
83 //mayor vida en el ArrayList
84 public static Soldado soldadoMayorVida (ArrayList<Soldado> array) {
85     int mayor=0;
86     Soldado soldadoMayor=null;
87     System.out.println(x:"\n_____");
88     for (int i=0;i<array.size();i++) {
89         int vida=array.get(i).getNivelVida();
90         if (vida>mayor) {
91             mayor=vida;
92             soldadoMayor=array.get(i);
93         }
94     }
95     return soldadoMayor;
}
```

```
96     }
97     //Metodo que retorna el promedio del nivel de vida de todos los soldados creados de
98     //un ejercito
99     public static double promedioNivelVida (ArrayList<Soldado> ejer){
100         double acumulado=0;
101         for (int i=0;i<ejer.size();i++){
102             double vida=ejer.get(i).getNivelVida();
103             acumulado+=vida;
104         }
105         double promedio=Math.round(acumulado/(ejer.size()));
106         return promedio;
107     }
108     //Metodo que ordena a los soldados en el orden que fueron creados
109     //y los imprime en tal orden
110     //--->Se emplea el algoritmo de Ordenamiento Burbuja
111     public static void imprimirOrdenCreacion(ArrayList<Soldado> ejer){
112         System.out.println(x:"\n_____");
113         System.out.println(x:"Orden segun su creacion: ");
114         for (int i=1;i<ejer.size();i++){
115             for (int j=0;j<ejer.size()-1;j++){
116                 String ejer_1=ejer.get(j).getNombre();
117                 String ejer_2=ejer.get(j+1).getNombre();
118                 if ((ejer_1.compareTo(ejer_2)>0){
119                     Soldado temp=ejer.get(j);
120                     ejer.set(j, ejer.get(j+1));
121                     ejer.set(j+1, temp);
122                 }
123             }
124         }
125         for (int k=0;k<ejer.size();k++){
126             System.out.println(ejer.get(k));
127         }
128     }
129     //Metodo que rankea a los soldados decrecientemente segun su nivel de vida
130     //y los imprime en tal orden
131     //--->Se emplea el algoritmo de Ordenamiento por Seleccion
132     public static void imprimirRankingVida(ArrayList<Soldado> ejer){
133         System.out.println(x:"\n_____");
134         System.out.println(x:"Ranking del nivel de vida:");
135         int n=0;
136         for (int i=0;i<ejer.size()-1;i++){
137             int mayor=i;
138             n++;
139             for (int j=n;j<ejer.size();j++){
140                 if (ejer.get(j).getNivelVida()>ejer.get(mayor).getNivelVida()){
141                     mayor=j;
142                 }
143             }
144         }
145     }
```

```
144         Soldado temp=ejer.get(mayor);
145         ejer.set(mayor, ejer.get(i));
146         ejer.set(i, temp);
147     }
148     for (int k=0;k<ejer.size();k++){
149         System.out.println(ejer.get(k));
150     }
151 }
152
153 //Metodo que compara, primero la vida total del ejercito, luego el promedio de vida de cada ejercito.
154 //Si en estos dos criterios ocurriese un empate, se compara la cantidad de soldados con vida maxima (5) y
155 //se decide el ganador
156 public static void decidirEjercitoGanador(ArrayList<Soldado> ejer1, ArrayList<Soldado> ejer2){
157     System.out.println(x:"\n_____");
158     int contador_ejer1=0;
159     int contador_ejer2=0;
160     int acumulado1=0;
161     int acumulado2=0;
162     for (int i=0;i<ejer1.size();i++){
163         acumulado1+=ejer1.get(i).getNivelVida();
164     }
165     for (int j=0;j<ejer2.size();j++){
166         acumulado2+=ejer2.get(j).getNivelVida();
167     }
168     if (acumulado1>acumulado2){
169         contador_ejer1++;
170     } if (acumulado2>acumulado1){
171         contador_ejer2++;
172     }
173
174     if (promedioNivelVida(ejer1)>promedioNivelVida(ejer2)){
175         contador_ejer1++;
176     } if (promedioNivelVida(ejer2)>promedioNivelVida(ejer1)){
177         contador_ejer2++;
178     }
179
180     if (contador_ejer1==contador_ejer2){
181         int contador1=0;
182         int contador2=0;
183         for (int s=0;s<ejer1.size();s++){
184             if (ejer1.get(s).getNivelVida()==5){
185                 contador1++;
186             }
187         }
188         for (int r=0;r<ejer2.size();r++){
189             if (ejer2.get(r).getNivelVida()==5){
190                 contador2++;
191             }
192         }
193     }
194 }
```

completed Java: Ready

```
191         }  
192     }  
193     if (contador1>contador2){  
194         contador_ejer1++;  
195     }  
196     if (contador2>contador1){  
197         contador_ejer2++;  
198     }  
199 }  
200 if (contador_ejer1>contador_ejer2){  
201     System.out.println(x:"El ejercito ganador fue el Ejercito 1!!!");  
202 } if (contador_ejer2>contador_ejer1) {  
203     System.out.println(x:"El ejercito ganador fue el Ejercito 2!!!");  
204 } if (contador_ejer1==contador_ejer2){  
205     System.out.println(x:"Hubo un empate");  
206 }  
207 }  
208 }  
209
```

c. Ejecución del código:

 Tablero de posiciones:

				2							
			2	1							
					1						
				2							
1	1				2						
			1				2				
			2	2							
			1								

 Los soldados con mayor vida son:

-Ejercito 1: -Nombre: Soldado2x1 -Posición: (9,4) -Nivel de Vida: 4
 -Ejercito 2: -Nombre: Soldado5x2 -Posición: (8,4) -Nivel de Vida: 4

 El promedio del nivel de vida del ejercito 1 es: 2.0

El promedio del nivel de vida del ejercito 2 es: 2.0

 Orden segun su creacion:

-Nombre: Soldado0x1 -Posición: (5,2) -Nivel de Vida: 1
 -Nombre: Soldado1x1 -Posición: (3,6) -Nivel de Vida: 1
 -Nombre: Soldado2x1 -Posición: (9,4) -Nivel de Vida: 4
 -Nombre: Soldado3x1 -Posición: (5,1) -Nivel de Vida: 2
 -Nombre: Soldado4x1 -Posición: (6,4) -Nivel de Vida: 1
 -Nombre: Soldado5x1 -Posición: (2,5) -Nivel de Vida: 1

 Orden segun su creacion:

-Nombre: Soldado0x2 -Posición: (5,6) -Nivel de Vida: 2
 -Nombre: Soldado1x2 -Posición: (1,5) -Nivel de Vida: 3
 -Nombre: Soldado2x2 -Posición: (8,3) -Nivel de Vida: 3
 -Nombre: Soldado3x2 -Posición: (4,5) -Nivel de Vida: 1
 -Nombre: Soldado4x2 -Posición: (2,4) -Nivel de Vida: 2
 -Nombre: Soldado5x2 -Posición: (8,4) -Nivel de Vida: 4
 -Nombre: Soldado6x2 -Posición: (6,8) -Nivel de Vida: 1

 Ranking del nivel de vida:

-Nombre: Soldado2x1 -Posición: (9,4) -Nivel de Vida: 4
 -Nombre: Soldado3x1 -Posición: (5,1) -Nivel de Vida: 2
 -Nombre: Soldado0x1 -Posición: (5,2) -Nivel de Vida: 1
 -Nombre: Soldado1x1 -Posición: (3,6) -Nivel de Vida: 1
 -Nombre: Soldado4x1 -Posición: (6,4) -Nivel de Vida: 1
 -Nombre: Soldado5x1 -Posición: (2,5) -Nivel de Vida: 1

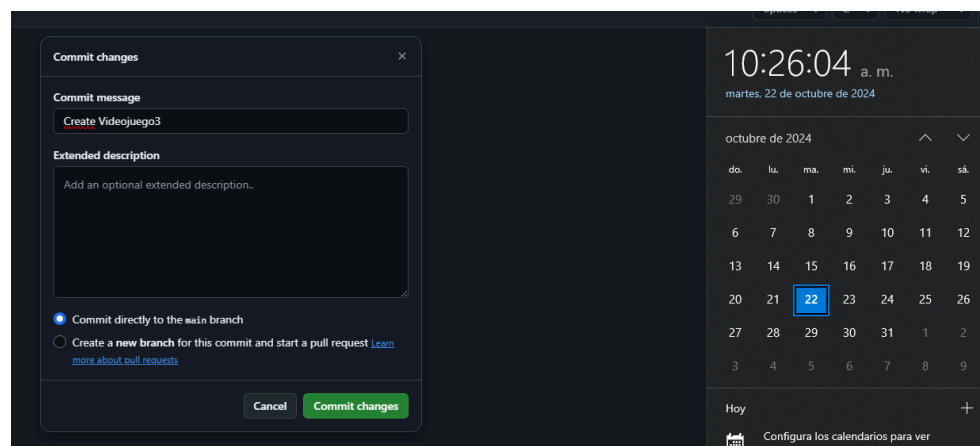
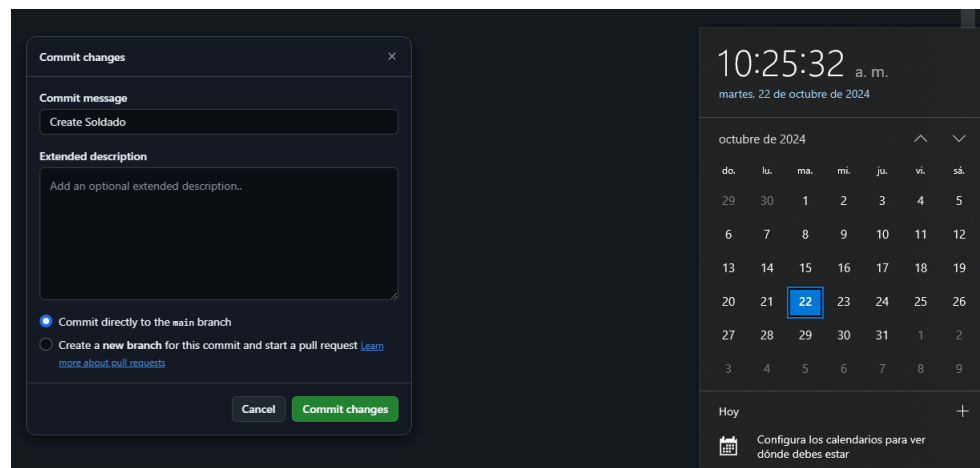

```

-----
Ranking del nivel de vida:
-Nombre: Soldado5x2      -Posición: (8,4)      -Nivel de Vida: 4
-Nombre: Soldado1x2      -Posición: (1,5)      -Nivel de Vida: 3
-Nombre: Soldado2x2      -Posición: (8,3)      -Nivel de Vida: 3
-Nombre: Soldado4x2      -Posición: (2,4)      -Nivel de Vida: 2
-Nombre: Soldado0x2      -Posición: (5,6)      -Nivel de Vida: 2
-Nombre: Soldado3x2      -Posición: (4,5)      -Nivel de Vida: 1
-Nombre: Soldado6x2      -Posición: (6,8)      -Nivel de Vida: 1

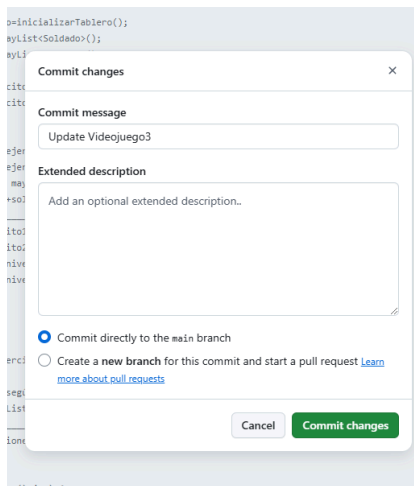
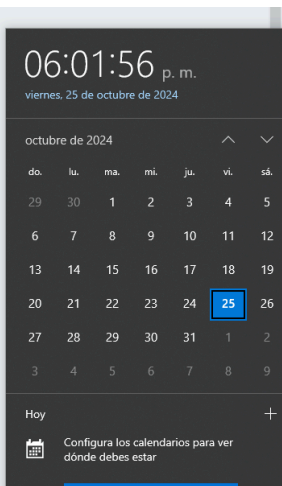
-----
El ejercito ganador fue el Ejercito 2!!!
PS C:\Users\Usuario248\Downloads\relacioness\labss>

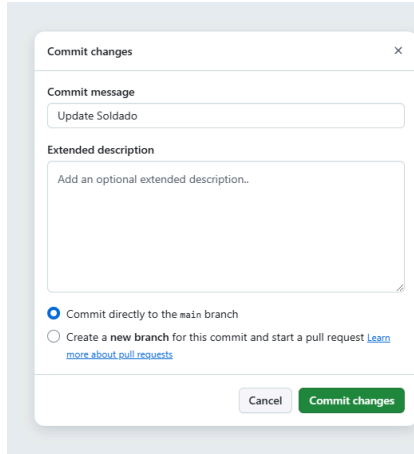
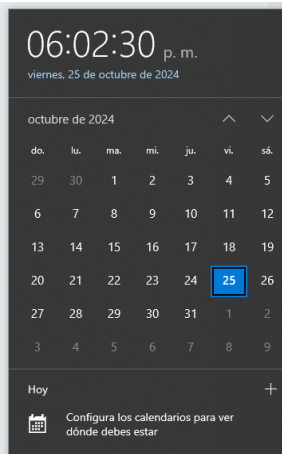
```

- **Evidencia de los Commits:**
 - **Commits del Laboratorio del dia 22 de octubre:**



○ **Commits del Laboratorio del día 25 de octubre:**

- **Link al repositorio de GitHub - Laboratorio05:**
https://github.com/SantiagoQuintanilla/LaboratorioFP2/tree/main/LABORATORIO_06
- **Autoevaluación:**

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 11

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		X	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2		X	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2		X	
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		X	
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			