

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: 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

(formato estudiante)

INFORMACIÓN BÁSICA					
ASIGNATURA:	Fundamentos de la programación 2 - Laboratorio				
TÍTULO DE LA PRÁCTICA:	ArrayList				
NÚMERO DE PRÁCTICA:	06	AÑO LECTIVO:	2024	NRO. SEMESTRE:	2024-B
FECHA DE PRESENTACIÓN	25/10/2024	HORA DE PRESENTACIÓN	11/59/00		
INTEGRANTE (s) Auccacusi Conde Brayan Carlos				NOTA (0-20)	Nota colocada por el docente
DOCENTE(s): Ing. Lino Pinto Oppe					

RESULTADOS Y PRUEBAS
<p>I. EJERCICIOS RESUELTOS:</p> <p><i>El estudiante coloca la evidencia de los ejercicios propuestos realizados en la sesión de laboratorio, en el tiempo o duración indicado por el docente.</i></p> <p><i>El docente debe colocar la retroalimentación por cada ejercicio que el estudiante/grupo ha presentado</i></p>

```
2 import java.util.*;
3 public class VideoJuego3 {
4     public static void main(String [] args){
5         Random rand = new Random();
6         ArrayList<ArrayList<Soldado>> campo = new ArrayList<>();
7         for (int i = 0; i < 10; i++){
8             campo.add(new ArrayList<Soldado>());
9             for(int j = 0; j < 10 ; j++){
10                 campo.get(i).add(null);
11             }
12         }
13         int i = 1;
14         while(i <= 20) {
15             int fila = rand.nextInt(10);
16             int columna = rand.nextInt(10);
17             while(campo.get(fila).get(columna) != null){
18                 fila = rand.nextInt(10);
19                 columna = rand.nextInt(10);
20             }
21             Soldado soldado = new Soldado();
22             if(i <= 10){
23                 if(i == 10){
24                     soldado.setNombre("Soldado10X1");
25                 } else {
26                     soldado.setNombre("Soldado0" + i + "X1");
27                 }
28             }
29
30             if(i > 10){
31                 if(i == 20){
32                     soldado.setNombre("Soldado10X2");
33                 } else {
34                     soldado.setNombre("Soldado0" + (i - 10 )+ "X2");
35                 }
36             }
37
38             soldado.setVida(rand.nextInt(5) + 1);
39             soldado.setFila(fila);
40             soldado.setColumna((char)('A' + columna));
41             soldado.setEstado(true);
42             campo.get(fila).set(columna, soldado);
43             i++;
44         }
45
46         mostrarTabla(campo);
47         mostrarMayorVida(campo);
48         promedioVida(campo);
49         System.out.println("\nDatos de los soldados en orden de creación:");
50         mostrar(campo);
51         aleatorio(campo);
52         System.out.println("\nAleatorio");
53         mostrar(campo);
54         System.out.println("\nRanking de mayor a menor BURBUJA");
55         rankingMayorMenorBurbuja(campo);
56         mostrar(campo);
57         aleatorio(campo);
58         System.out.println("\nAleatorio");
59         mostrar(campo);
60         System.out.println("\nRanking de mayor a menor SELECCION ");
61         rankingMayorMenorSeleccion(campo);
62         mostrar(campo);
63         determinarGanador(campo);
64     }
65 }
```

```
65
66
67 public static void mostrarTabla(ArrayList<ArrayList<Soldado>> campo){
68     for(char i = 'A'; i < 'K'; i++){
69         System.out.print("      " + i + "  ");
70     }
71
72     System.out.println();
73     for(int i = 0; i < 10; i++){
74         if(i == 9){
75             System.out.print(i + 1);
76         }
77         else
78             System.out.print((i + 1) + " ");
79
80         for(int j = 0; j < 10; j++){
81             if (campo.get(i).get(j) != null && campo.get(i).get(j).getEstado())
82                 System.out.print("|" + campo.get(i).get(j).getNombre());
83             else
84                 System.out.print("|          ");
85         }
86         System.out.println("|");
87         System.out.print("-----"+
88             "-----");
89     }
90 }
91
92
93 public static void mostrarMayorVida(ArrayList<ArrayList<Soldado>> campo){
94     Soldado mayor1 = new Soldado();
95     Soldado mayor2 = new Soldado();
96
97     mayor1.setVida(1);
98     mayor2.setVida(1);
99     int filMayorX1 = 0, colMayorX1 = 0, filMayorX2 = 0, colMayorX2 = 0;
100    // Recorremos todo el campo de batalla (array bidimensional)
101    for (int i = 0; i < campo.size(); i++){
102        for(int j = 0; j < campo.get(i).size(); j++){
103            // Verificamos que haya un soldado en la posición actual
104            if (campo.get(i).get(j) != null){
105                // Si el soldado pertenece al ejército 1 y tiene más vida que el actual mayor1
106                if(campo.get(i).get(j).getNombre().charAt(campo.get(i).get(j).getNombre().length() - 1) == '1' &&
107                    campo.get(i).get(j).getVida() > mayor1.getVida()){
108                    // Actualizamos el soldado mayor1 y su posición
109                    mayor1 = campo.get(i).get(j);
110                    filMayorX1 = i;
111                    colMayorX1 = j;
112                }
113                // Si el soldado pertenece al ejército 2 y tiene más vida que el actual mayor2
114                if(campo.get(i).get(j).getNombre().charAt(campo.get(i).get(j).getNombre().length() - 1) == '2' &&
115                    campo.get(i).get(j).getVida() > mayor2.getVida()){
116                    // Actualizamos el soldado mayor2 y su posición
117                    mayor2 = campo.get(i).get(j);
118                    filMayorX2 = i;
119                    colMayorX2 = j;
120                }
121            }
122        }
123    }
124
125    System.out.println("El soldado con mayor nivel de vida del ejercito X1 es: ");
126    System.out.println(campo.get(filMayorX1).get(colMayorX1).toString());
127    System.out.println("\nEl soldado con mayor nivel de vida del ejercito X2 es: ");
128    System.out.println(campo.get(filMayorX2).get(colMayorX2).toString());
129 }
130
```

```
130
131
132 public static void promedioVida(ArrayList<ArrayList<Soldado>> campo){
133     double sumEj1 = 0, sumEj2 = 0;
134     // Recorre todo el campo de soldados
135     for(int i = 0; i < campo.size(); i++){
136         for(int j = 0; j < campo.get(i).size(); j++){
137             // Si la posición actual contiene un soldado, suma su vida al total
138             if(campo.get(i).get(j) != null){
139                 if(campo.get(i).get(j).getNombre().charAt(campo.get(i).get(j).getNombre().length() - 1) == '1'){
140                     // Acumula el valor de la vida del soldado
141                     sumEj1 += campo.get(i).get(j).getVida();
142                 }
143                 if(campo.get(i).get(j).getNombre().charAt(campo.get(i).get(j).getNombre().length() - 1) == '2'){
144                     // Acumula el valor de la vida del soldado
145                     sumEj2 += campo.get(i).get(j).getVida();
146                 }
147             }
148         }
149     }
150 }
151 System.out.println("\nEl promedio de vida de los soldados del ejercito 1 es: " + (sumEj1 / 10));
152 System.out.println("El promedio de vida de los soldados del ejercito 2 es: " + (sumEj2 / 10));
153 }
154
155
156 public static void mostrar(ArrayList<ArrayList<Soldado>> campo){
157     System.out.println();
158     // Recorre todo el campo de soldados
159     System.out.println("Ejercito 1:");
160     for (int i = 0; i < campo.size(); i++){
161         for (int j = 0; j < campo.get(i).size(); j++){
162             // Si hay un soldado en la posición actual, imprime su información
163             if(campo.get(i).get(j) != null){
164                 if (campo.get(i).get(j).getNombre().charAt(campo.get(i).get(j).getNombre().length() - 1) == '1'){
165                     System.out.println(campo.get(i).get(j).toString());
166                 }
167             }
168         }
169     }
170     System.out.println("Ejercito 2:");
171     for (int i = 0; i < campo.size(); i++){
172         for (int j = 0; j < campo.get(i).size(); j++){
173             // Si hay un soldado en la posición actual, imprime su información
174             if(campo.get(i).get(j) != null){
175                 if (campo.get(i).get(j).getNombre().charAt(campo.get(i).get(j).getNombre().length() - 1) == '2'){
176                     System.out.println(campo.get(i).get(j).toString());
177                 }
178             }
179         }
180     }
181 }
```

```
182
183
184 public static void aleatorio(ArrayList<ArrayList<Soldado>> campo){
185     Random rand = new Random();
186     Soldado temp;
187     // Recorre todo el campoo de soldados
188     for (int i = 0; i < campo.size(); i++){
189         for (int j = 0; j < campo.get(i).size(); j++){
190             // Genera una fila y columna aleatoria dentro de la matriz
191             int randFila = rand.nextInt(campo.size());
192             int randColumna = rand.nextInt(campo.get(0).size());
193             // Intercambia el soldado en (i, j) con uno en una posición aleatoria (randFila, randColumna)
194             temp = campo.get(i).get(j);
195             campo.get(i).set(j, campo.get(randFila).get(randColumna));
196             campo.get(randFila).set(randColumna, temp);
197         }
198     }
199 }
200
201
202 public static void rankingMayorMenorBurbuja(ArrayList<ArrayList<Soldado>> campo) {
203     ArrayList<Soldado> ejercito1 = new ArrayList<>();
204     ArrayList<Soldado> ejercito2 = new ArrayList<>();
205
206     // Dividir soldados en Ejército 1 y Ejército 2
207     for (int i = 0; i < campo.size(); i++) {
208         for (int j = 0; j < campo.get(i).size(); j++) {
209             Soldado soldado = campo.get(i).get(j);
210             if (soldado != null) {
211                 // Si el nombre termina en '1', pertenece al Ejército 1
212                 if (soldado.getNombre().charAt(soldado.getNombre().length() - 1) == '1') {
213                     ejercito1.add(soldado);
214                 }
215                 // Si el nombre termina en '2', pertenece al Ejército 2
216                 else if (soldado.getNombre().charAt(soldado.getNombre().length() - 1) == '2') {
217                     ejercito2.add(soldado);
218                 }
219             }
220         }
221     }
222
223     // Ordenar Ejército 1 en orden descendente de vida (mayor a menor)
224     for (int i = 0; i < ejercito1.size() - 1; i++) {
225         for (int j = 0; j < ejercito1.size() - 1 - i; j++) {
226             if (ejercito1.get(j).getVida() < ejercito1.get(j + 1).getVida()) {
227                 // Intercambiar si está en el orden incorrecto
228                 Soldado temp = ejercito1.get(j);
229                 ejercito1.set(j, ejercito1.get(j + 1));
230                 ejercito1.set(j + 1, temp);
231             }
232         }
233     }
234
235     // Ordenar Ejército 2 en orden descendente de vida (mayor a menor)
236     for (int i = 0; i < ejercito2.size() - 1; i++) {
237         for (int j = 0; j < ejercito2.size() - 1 - i; j++) {
238             if (ejercito2.get(j).getVida() < ejercito2.get(j + 1).getVida()) {
239                 // Intercambiar si está en el orden incorrecto
240                 Soldado temp = ejercito2.get(j);
241                 ejercito2.set(j, ejercito2.get(j + 1));
242                 ejercito2.set(j + 1, temp);
243             }
244         }
245     }
246 }
247
248
```

```
249
250 public static void rankingMayorMenorSeleccion(ArrayList<ArrayList<Soldado>> campo){
251     Soldado temp1, temp2;
252     // Recorre todas las posiciones del campo
253     for (int i = 0; i < campo.size(); i++){
254         for (int j = 0; j < campo.get(i).size(); j++){
255             int filMenorEj1 = i, colMenorEj1 = j, filMenorEj2 = i, colMenorEj2 = j;
256
257             // Busca el soldado con la menor vida a partir de la posición actual
258             for (int k = i; k < campo.size(); k++){
259                 for (int l = 0; l < campo.get(k).size(); l++){
260
261                     // Evita revisar posiciones ya evaluadas
262                     if(campo.get(i).get(j) != null){
263                         if (campo.get(i).get(j).getNombre().charAt(campo.get(i).get(j).getNombre().length() - 1) == '1'){
264                             if (k == i && l <= j) {
265                                 continue;
266                             }
267                             // Cambia las coordenadas si se encuentra un soldado con vida más baja
268                             if (campo.get(k).get(l) != null && campo.get(filMenorEj1).get(colMenorEj1) != null){
269                                 if (campo.get(k).get(l).getVida() < campo.get(filMenorEj1).get(colMenorEj1).getVida()){
270                                     filMenorEj1 = k;
271                                     colMenorEj1 = l;
272                                 }
273                             }
274                         }
275                         if (campo.get(i).get(j).getNombre().charAt(campo.get(i).get(j).getNombre().length() - 1) == '2'){
276                             if (k == i && l <= j) {
277                                 continue;
278                             }
279                             // Cambia las coordenadas si se encuentra un soldado con vida más baja
280                             if (campo.get(k).get(l) != null && campo.get(filMenorEj2).get(colMenorEj2) != null){
281                                 if (campo.get(k).get(l).getVida() > campo.get(filMenorEj2).get(colMenorEj2).getVida()){
282                                     filMenorEj2 = k;
283                                     colMenorEj2 = l;
284                                 }
285                             }
286                         }
287                     }
288                 }
289             }
290         }
291         // Intercambia los soldados si se encontró uno con menor vida
292         if (filMenorEj1 != i || colMenorEj1 != j){
293             temp1 = campo.get(i).get(j);
294             campo.get(i).set(j, campo.get(filMenorEj1).get(colMenorEj1));
295             campo.get(filMenorEj1).set(colMenorEj1, temp1);
296         }
297         if (filMenorEj2 != i || colMenorEj2 != j){
298             temp2 = campo.get(i).get(j);
299             campo.get(i).set(j, campo.get(filMenorEj2).get(colMenorEj2));
300             campo.get(filMenorEj2).set(colMenorEj2, temp2);
301         }
302     }
303 }
304 }
305 }
```


Datos de los soldados en orden de creación:**Ejercito 1:**

Nombre: Soldado04X1	Vida: 2	Fila 2	Columna: F
Nombre: Soldado08X1	Vida: 3	Fila 3	Columna: F
Nombre: Soldado07X1	Vida: 5	Fila 3	Columna: J
Nombre: Soldado02X1	Vida: 5	Fila 4	Columna: B
Nombre: Soldado03X1	Vida: 1	Fila 4	Columna: D
Nombre: Soldado09X1	Vida: 4	Fila 5	Columna: J
Nombre: Soldado01X1	Vida: 1	Fila 6	Columna: J
Nombre: Soldado06X1	Vida: 1	Fila 8	Columna: J
Nombre: Soldado10X1	Vida: 5	Fila 9	Columna: D
Nombre: Soldado05X1	Vida: 1	Fila 10	Columna: F

Ejercito 2:

Nombre: Soldado04X2	Vida: 5	Fila 1	Columna: B
Nombre: Soldado06X2	Vida: 2	Fila 2	Columna: B
Nombre: Soldado08X2	Vida: 3	Fila 2	Columna: H
Nombre: Soldado02X2	Vida: 5	Fila 3	Columna: G
Nombre: Soldado01X2	Vida: 5	Fila 3	Columna: I
Nombre: Soldado07X2	Vida: 4	Fila 5	Columna: H
Nombre: Soldado09X2	Vida: 2	Fila 8	Columna: C
Nombre: Soldado10X2	Vida: 4	Fila 9	Columna: C
Nombre: Soldado03X2	Vida: 2	Fila 10	Columna: E
Nombre: Soldado05X2	Vida: 2	Fila 10	Columna: J

Aleatorio**Ejercito 1:**

Nombre: Soldado10X1	Vida: 5	Fila 9	Columna: D
Nombre: Soldado08X1	Vida: 3	Fila 3	Columna: F
Nombre: Soldado07X1	Vida: 5	Fila 3	Columna: J
Nombre: Soldado09X1	Vida: 4	Fila 5	Columna: J
Nombre: Soldado03X1	Vida: 1	Fila 4	Columna: D
Nombre: Soldado02X1	Vida: 5	Fila 4	Columna: B
Nombre: Soldado05X1	Vida: 1	Fila 10	Columna: F
Nombre: Soldado04X1	Vida: 2	Fila 2	Columna: F
Nombre: Soldado06X1	Vida: 1	Fila 8	Columna: J
Nombre: Soldado01X1	Vida: 1	Fila 6	Columna: J

Ejercito 2:

Nombre: Soldado05X2	Vida: 2	Fila 10	Columna: J
Nombre: Soldado04X2	Vida: 5	Fila 1	Columna: B
Nombre: Soldado10X2	Vida: 4	Fila 9	Columna: C
Nombre: Soldado08X2	Vida: 3	Fila 2	Columna: H
Nombre: Soldado07X2	Vida: 4	Fila 5	Columna: H
Nombre: Soldado09X2	Vida: 2	Fila 8	Columna: C
Nombre: Soldado02X2	Vida: 5	Fila 3	Columna: G
Nombre: Soldado01X2	Vida: 5	Fila 3	Columna: I
Nombre: Soldado06X2	Vida: 2	Fila 2	Columna: B
Nombre: Soldado03X2	Vida: 2	Fila 10	Columna: E

Ranking de mayor a menor BURBUJA

Ejercito 1:

Nombre: Soldado10X1	Vida: 5	Fila 9	Columna: D
Nombre: Soldado08X1	Vida: 3	Fila 3	Columna: F
Nombre: Soldado07X1	Vida: 5	Fila 3	Columna: J
Nombre: Soldado09X1	Vida: 4	Fila 5	Columna: J
Nombre: Soldado03X1	Vida: 1	Fila 4	Columna: D
Nombre: Soldado02X1	Vida: 5	Fila 4	Columna: B
Nombre: Soldado05X1	Vida: 1	Fila 10	Columna: F
Nombre: Soldado04X1	Vida: 2	Fila 2	Columna: F
Nombre: Soldado06X1	Vida: 1	Fila 8	Columna: J
Nombre: Soldado01X1	Vida: 1	Fila 6	Columna: J

Ejercito 2:

Nombre: Soldado05X2	Vida: 2	Fila 10	Columna: J
Nombre: Soldado04X2	Vida: 5	Fila 1	Columna: B
Nombre: Soldado10X2	Vida: 4	Fila 9	Columna: C
Nombre: Soldado08X2	Vida: 3	Fila 2	Columna: H
Nombre: Soldado07X2	Vida: 4	Fila 5	Columna: H
Nombre: Soldado09X2	Vida: 2	Fila 8	Columna: C
Nombre: Soldado02X2	Vida: 5	Fila 3	Columna: G
Nombre: Soldado01X2	Vida: 5	Fila 3	Columna: I
Nombre: Soldado06X2	Vida: 2	Fila 2	Columna: B
Nombre: Soldado03X2	Vida: 2	Fila 10	Columna: E

Aleatorio

Ejercito 1:

Nombre: Soldado07X1	Vida: 5	Fila 3	Columna: J
Nombre: Soldado08X1	Vida: 3	Fila 3	Columna: F
Nombre: Soldado10X1	Vida: 5	Fila 9	Columna: D
Nombre: Soldado03X1	Vida: 1	Fila 4	Columna: D
Nombre: Soldado09X1	Vida: 4	Fila 5	Columna: J
Nombre: Soldado05X1	Vida: 1	Fila 10	Columna: F
Nombre: Soldado01X1	Vida: 1	Fila 6	Columna: J
Nombre: Soldado06X1	Vida: 1	Fila 8	Columna: J
Nombre: Soldado02X1	Vida: 5	Fila 4	Columna: B
Nombre: Soldado04X1	Vida: 2	Fila 2	Columna: F

Ejercito 2:

Nombre: Soldado07X2	Vida: 4	Fila 5	Columna: H
Nombre: Soldado02X2	Vida: 5	Fila 3	Columna: G
Nombre: Soldado05X2	Vida: 2	Fila 10	Columna: J
Nombre: Soldado06X2	Vida: 2	Fila 2	Columna: B
Nombre: Soldado01X2	Vida: 5	Fila 3	Columna: I
Nombre: Soldado03X2	Vida: 2	Fila 10	Columna: E
Nombre: Soldado09X2	Vida: 2	Fila 8	Columna: C
Nombre: Soldado10X2	Vida: 4	Fila 9	Columna: C
Nombre: Soldado08X2	Vida: 3	Fila 2	Columna: H
Nombre: Soldado04X2	Vida: 5	Fila 1	Columna: B

Ranking de mayor a menor SELECCION

Ejercito 1:

Nombre: Soldado07X1	Vida: 5	Fila 3	Columna: J
Nombre: Soldado10X1	Vida: 5	Fila 9	Columna: D
Nombre: Soldado03X1	Vida: 1	Fila 4	Columna: D
Nombre: Soldado02X1	Vida: 5	Fila 4	Columna: B
Nombre: Soldado05X1	Vida: 1	Fila 10	Columna: F
Nombre: Soldado01X1	Vida: 1	Fila 6	Columna: J
Nombre: Soldado06X1	Vida: 1	Fila 8	Columna: J
Nombre: Soldado09X1	Vida: 4	Fila 5	Columna: J
Nombre: Soldado08X1	Vida: 3	Fila 3	Columna: F
Nombre: Soldado04X1	Vida: 2	Fila 2	Columna: F

Ejercito 2:

Nombre: Soldado02X2	Vida: 5	Fila 3	Columna: G
Nombre: Soldado01X2	Vida: 5	Fila 3	Columna: I
Nombre: Soldado04X2	Vida: 5	Fila 1	Columna: B
Nombre: Soldado10X2	Vida: 4	Fila 9	Columna: C
Nombre: Soldado07X2	Vida: 4	Fila 5	Columna: H
Nombre: Soldado08X2	Vida: 3	Fila 2	Columna: H
Nombre: Soldado09X2	Vida: 2	Fila 8	Columna: C
Nombre: Soldado03X2	Vida: 2	Fila 10	Columna: E
Nombre: Soldado05X2	Vida: 2	Fila 10	Columna: J
Nombre: Soldado06X2	Vida: 2	Fila 2	Columna: B

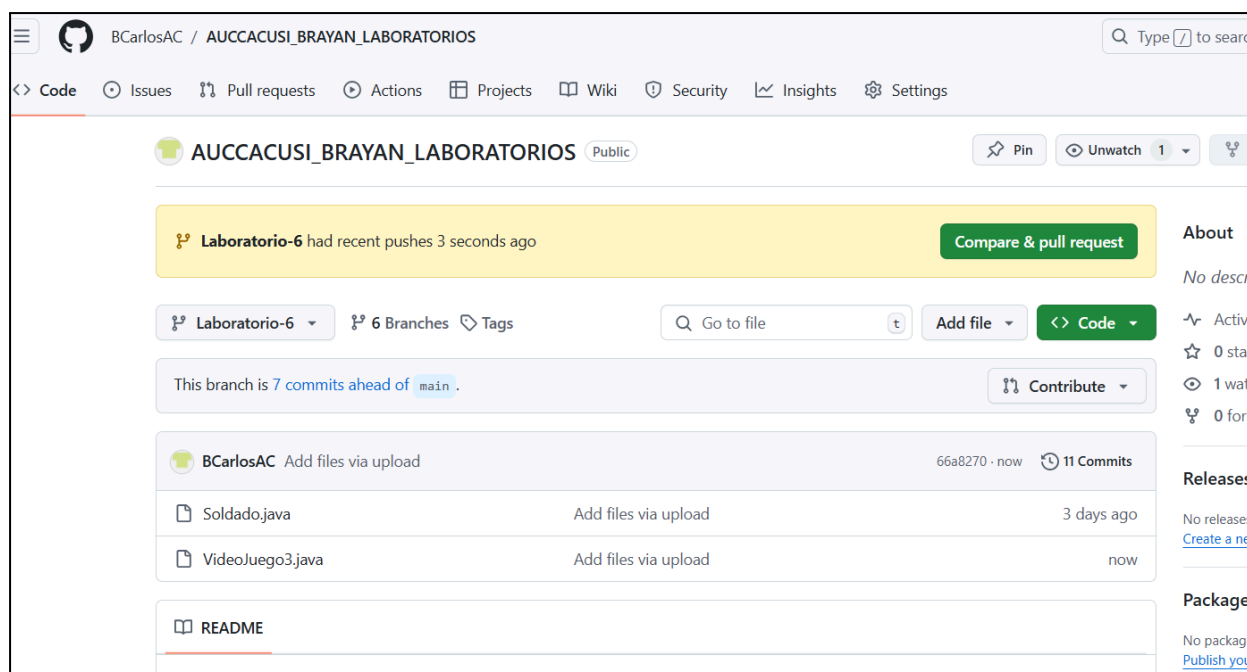
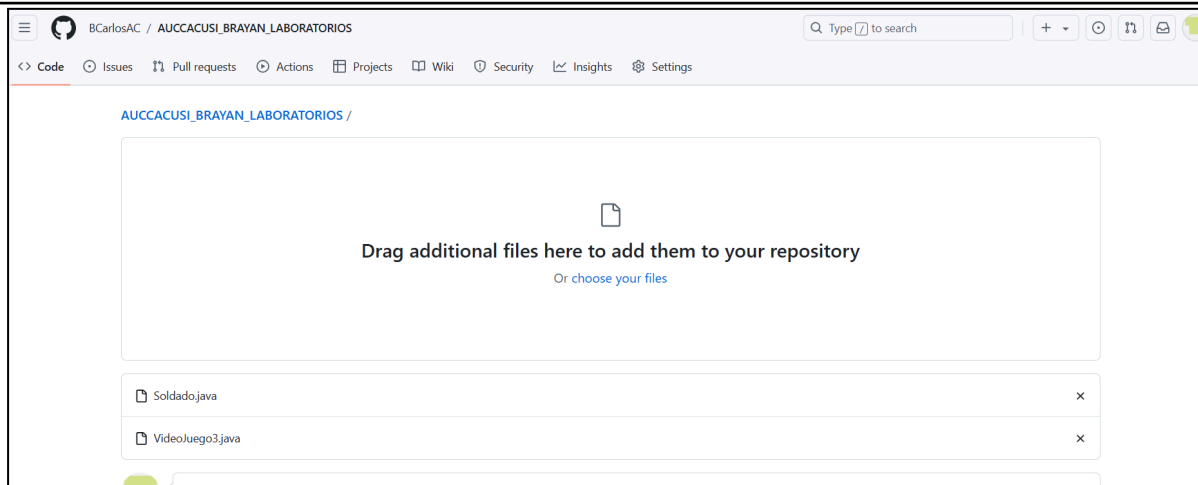
GANA EL EJERCITO X2

Puntos de vida totales:

Ejercito X1: 28.0 < 34.0 Ejercito X2

PS C:\Users\Hogar\Documents\BRAYAN\FP2 - Laboratories>

	<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: 11



¿Con que valores comprobaste que tu práctica estuviera correcta? ¿Qué resultado esperabas obtener para cada valor de entrada? ¿Qué valor o comportamiento obtuviste para cada valor de entrada?

- *Solo inicialice mi código debido a que los valores se determinaban aleatoriamente.*
- *Esperaba obtener una tabla y una especie de formato de mis métodos.*

III. CUESTIONARIO:

Colocar la evidencia de las respuestas realizadas al cuestionario enunciado en la guía práctica de laboratorio.

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	2	
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	2	
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	2	
TOTAL		20		18	

CONCLUSIONES

Los arraylist son una forma eficaz de trabajar ya que no ahorran recurso debido a que se autodimensionan.

METODOLOGÍA DE TRABAJO

Copie mi código anterior.

Verifique lo que me pedía el ejercicio.

Reemplace lo máximo que pude

Hice prueba y error para verificar el output, buscando información

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 13</p>

REFERENCIAS Y BIBLIOGRAFÍA
<p><i>Colocare las referencias utilizadas para el desarrollo de la práctica en formato IEEE</i></p>