



## CLASE SOLDADO

```
1 package L7;
2
3 public class Soldado {
4     private String nombre;
5     private int vida;
6     private int fila;
7     private int columna;
8
9     //Metodos mutadores
10    public void setNombre( String n){
11        nombre = n;
12    }
13    public void setFila(int f){
14        fila = f;
15    }
16    public void setColumna(int c){
17        columna = c;
18    }
19    public void setVida(int v){
20        vida = v;
21    }
22
23    // Metodos accesoros
24    public String getNombre(){
25        return nombre;
26    }
27    public int getFila(){
28        return fila;
29    }
30    public int getColumna(){
31        return columna;
32    }
33    public int getVida(){
34        return vida;
35    }
36
37    public String toString(){
38        return "Nombre: " + getNombre() + "      Vida: " + getVida() +
39            "      Fila " + (getFila() + 1) + "      Columna: " + getColumna();
40    }
41 }
```

## CLASE VIDEOJUEGO4

```
1 package L7;
2 import java.util.*;
3 public class VideoJuego4 {
4     public static void main(String [] args){
5         char decicion = 'y';
6         while(decicion == 'y'){
7             Scanner sc = new Scanner(System.in);
8             Random rand = new Random();
9             Soldado[][] campo = new Soldado[10][10];
10            Soldado[] ej1 = new Soldado[rand.nextInt(10) + 1];
11            Soldado[] ej2 = new Soldado[rand.nextInt(10) + 1];
12            crearSoldados(ej1, '@');
13            crearSoldados(ej2, '?');
14            asignarSoldados(campo, ej1);
15            asignarSoldados(campo, ej2);
16            mostrarTabla(campo);
17            System.out.println("Mayor vida ejercito 1: \n\t" + ej1[mayorVida(ej1)]);
18            System.out.println("Mayor vida ejercito 2: \n\t" + ej2[mayorVida(ej2)]);
19            System.out.println("\nPromedio vida ejercito 1: " + promedioEjercito(ej1));
20            System.out.println("Promedio vida ejercito 2: " + promedioEjercito(ej2));
21            System.out.println("\nDatos en orden que fueron creados");
22            mostrar(ej1, 1);
23            mostrar(ej2, 2);
24            System.out.println("\nRanking de poder mayor a menor vida");
25            System.out.println("BURBUJA");
26            rankingMayorMenorBurbuja(ej1);
27            mostrar(ej1, 1);
28            rankingMayorMenorBurbuja(ej2);
29            mostrar(ej2, 2);
30
31            System.out.println("\nAleatorio");
32            aleatorio(ej1);
33            aleatorio(ej2);
34            mostrar(ej1, 1);
35            mostrar(ej2, 2);
36
37            System.out.println("\nSELECCION");
38            rankingMayorMenorSeleccion(ej1);
39            mostrar(ej1, 1);
40            rankingMayorMenorSeleccion(ej2);
41            mostrar(ej2, 2);
42
43            System.out.println();
44            determinarGanador(ej1, ej2);
45
46            System.out.print("Desea iniciar un nuevo juego? (y/n): ");
47            decicion = sc.nextLine().charAt(0);
48        }
49    }
50 }
```

```
51 public static void crearSoldados(Soldado[] ej, char a){
52     Random rand = new Random();
53     for(int i = 0; i < ej.length; i++){
54         ej[i] = new Soldado();
55         ej[i].setVida(rand.nextInt(5) + 1);
56         ej[i].setNombre(i + "." + a + ej[i].getVida());
57     }
58 }
59 public static void asignarSoldados(Soldado[][] campo, Soldado[] ej){
60     Random rand = new Random();
61     int fila, columna;
62     for (int i = 0; i < ej.length; i++){
63         boolean aux = true;
64         while(aux){
65             fila = rand.nextInt(10);
66             columna = rand.nextInt(10);
67             if(campo[fila][columna] == null){
68                 campo[fila][columna] = ej[i];
69                 ej[i].setFila(fila);
70                 ej[i].setColumna(columna);
71                 aux = false;
72             }
73         }
74     }
75 }
76 }
77 public static void mostrarTabla(Soldado[][] campo){
78     System.out.print(" ");
79     for(char i = 'A'; i < 'K'; i++){
80         System.out.print(" " + i + " ");
81     }
82     System.out.println();
83     for(int i = 0; i < campo.length; i++){
84         if(i == 9)
85             System.out.print(i + 1);
86         else
87             System.out.print((i + 1) + " ");
88
89         for(int j = 0; j < campo[i].length; j++){
90             if (campo[i][j] != null)
91                 System.out.print("|" + campo[i][j].getNombre());
92             else
93                 System.out.print("|   ");
94         }
95         System.out.println(" |");
96         System.out.print(" -----");
97     }
98 }
```

```
99 public static int mayorVida(Soldado[] ej){
100     int indexMayor = 0;
101     for(int i = 0; i < ej.length - 1; i++){
102         if(ej[indexMayor].getVida() < ej[i].getVida()){
103             indexMayor = i;
104         }
105     }
106     return indexMayor;
107 }
108
109 public static double promedioEjercito(Soldado[] ej){
110     double sumLife = 0;
111     for(int i = 0; i < ej.length; i++){
112         sumLife += ej[i].getVida();
113     }
114     return sumLife/ej.length;
115 }
116
117 //Muestra el orden de creacion de los soldados, esto porque al asignar soldados al array
118 //lo hicimos de manera ordenada, ademas puede reutilizarse en otros metodos.
119 public static void mostrar(Soldado[] ej, int a){
120     System.out.println("Ejercito " + a);
121     for(Soldado sold : ej){
122         System.out.println(sold);
123     }
124 }
125
126 public static void aleatorio(Soldado[] ej){
127     Random rand = new Random();
128     Soldado aux;
129     int r1, r2;
130     for(int i = 0; i < ej.length; i++){
131         r1 = rand.nextInt(ej.length);
132         r2 = rand.nextInt(ej.length);
133         aux = ej[r1];
134         ej[r1] = ej[r2];
135         ej[r2] = aux;
136     }
137 }
138
139 public static void rankingMayorMenorBurbuja(Soldado[] ej){
140     for (int i = 0; i < ej.length - 1; i++){
141         for(int j = 0; j < ej.length - i - 1; j++){
142             if(ej[j].getVida() < ej[j + 1].getVida()){
143                 Soldado aux = ej[j];
144                 ej[j] = ej[j + 1];
145                 ej[j + 1] = aux;
146             }
147         }
148     }
149 }
```

```
150
151     public static void rankingMayorMenorSeleccion(Soldado[] ej){
152         for (int i = 0; i < ej.length - 1; i++){
153             int indexMayor = i;
154             for (int j = i + 1; j < ej.length; j++){
155                 if(ej[j].getVida() > ej[indexMayor].getVida())
156                     indexMayor = j;
157             }
158             Soldado aux = ej[i];
159             ej[i] = ej[indexMayor];
160             ej[indexMayor] = aux;
161         }
162     }
163
164     public static void determinarGanador(Soldado[] ej1, Soldado[] ej2){
165         double sumLifeEj1 = 0, sumLifeEj2 = 0;
166         for(int i = 0; i < ej1.length; i++){
167             sumLifeEj1 += ej1[i].getVida();
168         }
169         for(int i = 0; i < ej2.length; i++){
170             sumLifeEj2 += ej2[i].getVida();
171         }
172         if (sumLifeEj1 == sumLifeEj2) {
173             System.out.println("EMPATE\nEjercito1 = " + sumLifeEj1 +
174                 " ---- Ejercito2 = " + sumLifeEj2);
175         }
176         else if (sumLifeEj1 > sumLifeEj2) {
177             System.out.println("GANA EJERCITO1\nEjercito1 = " + sumLifeEj1 +
178                 " > Ejercito2 = " + sumLifeEj2);
179         }
180         else{
181             System.out.println("GANA EJERCITO2\nEjercito1 = " + sumLifeEj1 +
182                 " < Ejercito2 = " + sumLifeEj2);
183         }
184     }
185 }
```

## II. PRUEBAS

```

:Users\Hogar\Documents\BRAYAN\FP2 - Laboratories\bin' 'L7.VideoJ
  A B C D E F G H I J
1 | | | |6.?5| | | | | |
  -----
2 | | | | | |5.?4| | | | |
  -----
3 |0.?1| | | | | | | |3.?3|
  -----
4 |1.@3| | | | | | | |0.@3|
  -----
5 | | | | | | | |5.@4| | |
  -----
6 | | | | | | | |4.?4| | |
  -----
7 | | |2.@3| | | |3.@1|4.@2| | |
  -----
8 | |2.?2| |8.@4| | |7.@4| |6.@1| |
  -----
9 | | | |1.?3| | | | | | |
  -----
10| | | | | | | | | | |
  -----
Mayor vida ejercito 1:
      Nombre: 5.@4 Vida: 4 Filas 5 Columna: 7
Mayor vida ejercito 2:
      Nombre: 4.?4 Vida: 4 Filas 6 Columna: 8

Promedio vida ejercito 1: 2.7777777777777777
Promedio vida ejercito 2: 2.7777777777777777

```

```

Datos en orden que fueron creados
Ejercito 1
Nombre: 0.@3 Vida: 3 Filas 4 Columna: 9
Nombre: 1.@3 Vida: 3 Filas 4 Columna: 0
Nombre: 2.@3 Vida: 3 Filas 7 Columna: 2
Nombre: 3.@1 Vida: 1 Filas 7 Columna: 6
Nombre: 4.@2 Vida: 2 Filas 7 Columna: 7
Nombre: 5.@4 Vida: 4 Filas 5 Columna: 7
Nombre: 6.@1 Vida: 1 Filas 8 Columna: 8
Nombre: 7.@4 Vida: 4 Filas 8 Columna: 6
Nombre: 8.@4 Vida: 4 Filas 8 Columna: 3
Ejercito 2
Nombre: 0.?1 Vida: 1 Filas 3 Columna: 0
Nombre: 1.?3 Vida: 3 Filas 9 Columna: 3
Nombre: 2.?2 Vida: 2 Filas 8 Columna: 1
Nombre: 3.?3 Vida: 3 Filas 3 Columna: 9
Nombre: 4.?4 Vida: 4 Filas 6 Columna: 8
Nombre: 5.?4 Vida: 4 Filas 2 Columna: 5
Nombre: 6.?5 Vida: 5 Filas 1 Columna: 3

```

## Ranking de poder mayor a menor vida

## BURBUJA

## Ejercito 1

Nombre: 5.@4	Vida: 4	Fila 5	Columna: 7
Nombre: 7.@4	Vida: 4	Fila 8	Columna: 6
Nombre: 8.@4	Vida: 4	Fila 8	Columna: 3
Nombre: 0.@3	Vida: 3	Fila 4	Columna: 9
Nombre: 1.@3	Vida: 3	Fila 4	Columna: 0
Nombre: 2.@3	Vida: 3	Fila 7	Columna: 2
Nombre: 4.@2	Vida: 2	Fila 7	Columna: 7
Nombre: 3.@1	Vida: 1	Fila 7	Columna: 6
Nombre: 6.@1	Vida: 1	Fila 8	Columna: 8

## Ejercito 2

Nombre: 6.?5	Vida: 5	Fila 1	Columna: 3
Nombre: 4.?4	Vida: 4	Fila 6	Columna: 8
Nombre: 5.?4	Vida: 4	Fila 2	Columna: 5
Nombre: 1.?3	Vida: 3	Fila 9	Columna: 3
Nombre: 3.?3	Vida: 3	Fila 3	Columna: 9
Nombre: 2.?2	Vida: 2	Fila 8	Columna: 1
Nombre: 0.?1	Vida: 1	Fila 3	Columna: 0

## Aleatorio

## Ejercito 1

Nombre: 5.@4	Vida: 4	Fila 5	Columna: 7
Nombre: 3.@1	Vida: 1	Fila 7	Columna: 6
Nombre: 8.@4	Vida: 4	Fila 8	Columna: 3
Nombre: 1.@3	Vida: 3	Fila 4	Columna: 0
Nombre: 2.@3	Vida: 3	Fila 7	Columna: 2
Nombre: 4.@2	Vida: 2	Fila 7	Columna: 7
Nombre: 0.@3	Vida: 3	Fila 4	Columna: 9
Nombre: 7.@4	Vida: 4	Fila 8	Columna: 6
Nombre: 6.@1	Vida: 1	Fila 8	Columna: 8

## Ejercito 2

Nombre: 3.?3	Vida: 3	Fila 3	Columna: 9
Nombre: 4.?4	Vida: 4	Fila 6	Columna: 8
Nombre: 5.?4	Vida: 4	Fila 2	Columna: 5
Nombre: 2.?2	Vida: 2	Fila 8	Columna: 1
Nombre: 0.?1	Vida: 1	Fila 3	Columna: 0
Nombre: 1.?3	Vida: 3	Fila 9	Columna: 3
Nombre: 6.?5	Vida: 5	Fila 1	Columna: 3



**SELECCION****Ejercito 1**

Nombre: 5.@4	Vida: 4	Fila 5	Columna: 7
Nombre: 8.@4	Vida: 4	Fila 8	Columna: 3
Nombre: 7.@4	Vida: 4	Fila 8	Columna: 6
Nombre: 1.@3	Vida: 3	Fila 4	Columna: 0
Nombre: 2.@3	Vida: 3	Fila 7	Columna: 2
Nombre: 0.@3	Vida: 3	Fila 4	Columna: 9
Nombre: 4.@2	Vida: 2	Fila 7	Columna: 7
Nombre: 3.@1	Vida: 1	Fila 7	Columna: 6
Nombre: 6.@1	Vida: 1	Fila 8	Columna: 8

**Ejercito 2**

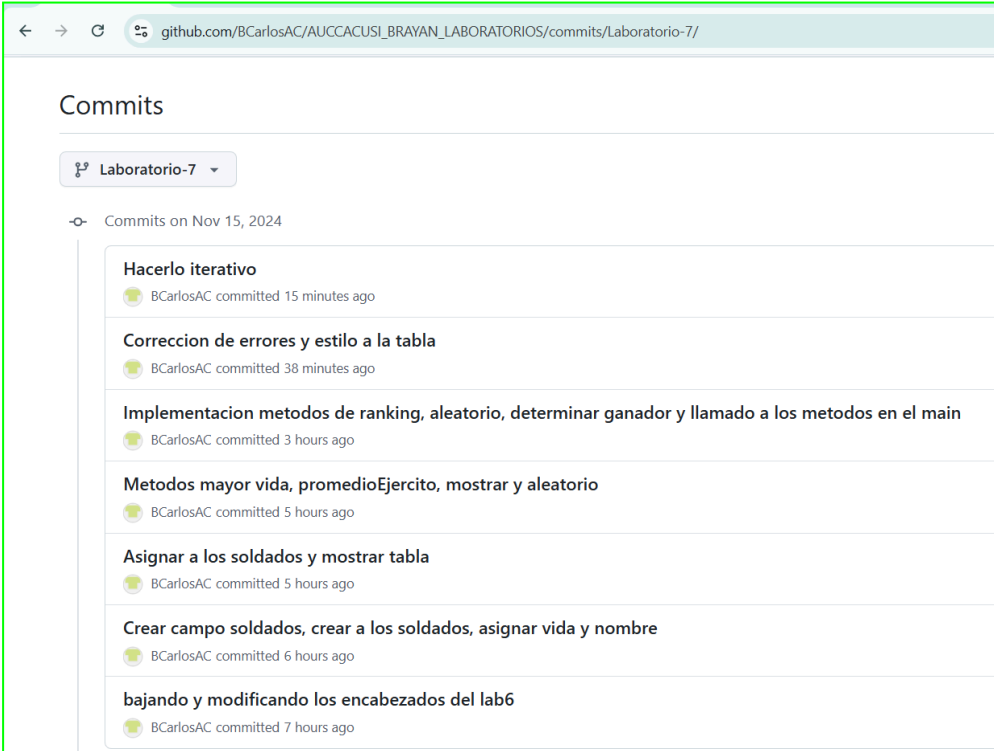
Nombre: 6.?5	Vida: 5	Fila 1	Columna: 3
Nombre: 4.?4	Vida: 4	Fila 6	Columna: 8
Nombre: 5.?4	Vida: 4	Fila 2	Columna: 5
Nombre: 1.?3	Vida: 3	Fila 9	Columna: 3
Nombre: 3.?3	Vida: 3	Fila 3	Columna: 9
Nombre: 2.?2	Vida: 2	Fila 8	Columna: 1
Nombre: 0.?1	Vida: 1	Fila 3	Columna: 0

**GANAR EJERCITO1****Ejercito1 = 25.0 > Ejercito2 = 22.0****Desea iniciar un nuevo juego? (y/n): n****PS C:\Users\Hogar\Documents\BRAYAN\FP2 - Laboratories>**

*¿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?*

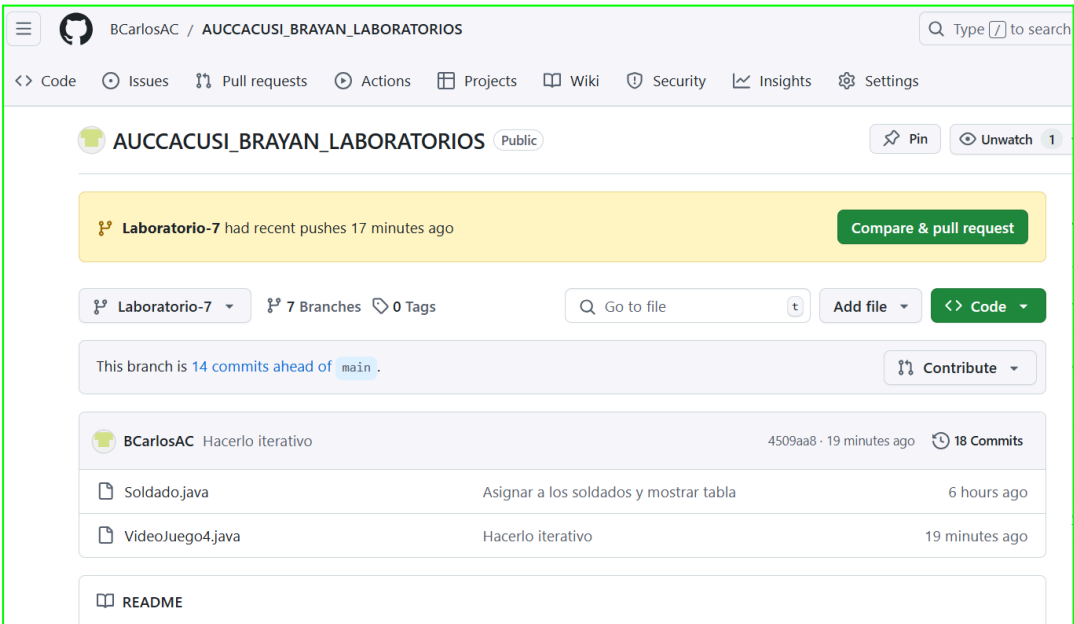
- *Ejecute y revise si había errores.*
- *La mayor parte del código se ejecuta automáticamente, no requiere mucha interacción con el usuario, solo en decidir si iniciar un nuevo juego.*
- *Al decidir que si se iniciaba un nuevo juego, si ponía no el programa finalizaba.*

## COMMITTS



The screenshot shows the GitHub Commits page for the repository `BCarlosAC/AUCCACUSI_BRAYAN_LABORATORIOS`, specifically for the branch `Laboratorio-7`. The page lists several commits made by `BCarlosAC` on November 15, 2024. The commits are as follows:

- Hacerlo iterativo** (15 minutes ago)
- Correccion de errores y estilo a la tabla** (38 minutes ago)
- Implementacion metodos de ranking, aleatorio, determinar ganador y llamado a los metodos en el main** (3 hours ago)
- Metodos mayor vida, promedioEjercito, mostrar y aleatorio** (5 hours ago)
- Asignar a los soldados y mostrar tabla** (5 hours ago)
- Crear campo soldados, crear a los soldados, asignar vida y nombre** (6 hours ago)
- bajando y modificando los encabezados del lab6** (7 hours ago)



The screenshot shows the GitHub repository page for `AUCCACUSI_BRAYAN_LABORATORIOS`. The repository is public and has 7 branches and 0 tags. The current branch is `Laboratorio-7`, which is 14 commits ahead of the `main` branch. The page shows the commit history for the `Laboratorio-7` branch, with the most recent commit being `Hacerlo iterativo` by `BCarlosAC` 19 minutes ago. The commit message is `4509aa8 · 19 minutes ago`. The page also shows the commit history for the `main` branch, with the most recent commit being `Asignar a los soldados y mostrar tabla` by `BCarlosAC` 6 hours ago. The page also shows the commit history for the `main` branch, with the most recent commit being `Hacerlo iterativo` by `BCarlosAC` 19 minutes ago.

**Enlace a mi repositorio:**

[https://github.com/BCarlosAC/AUCCACUSI\\_BRAYAN\\_LABORATORIOS.git](https://github.com/BCarlosAC/AUCCACUSI_BRAYAN_LABORATORIOS.git)

### III. CUESTIONARIO:

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	1	
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	7	18	

	<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>	
<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> 12

## CONCLUSIONES

*El array de objetos es una herramienta poderosa ya que podemos almacenar diferentes tipos de datos, como String, int, etc. Esto no ahorra el crear diferentes arrays para cada tipo de dato además de que nos permite trabajar ordenadamente.*

*Para trabajar con un arreglo de objetos, debemos crear la clase de esos objetos, además hay que considerar que los arreglos de objetos tienen una sintaxis más avanzada (debemos especificar un poco más a la hora de trabajar)*

*Ejemplo en un array normal basta con colocar el nombre del array + el índice para obtener un dato, en un array de objetos debemos colocar además de lo anterior los métodos creados en la clase, como los getter o setters..*

## METODOLOGÍA DE TRABAJO

1. *Primero leí y traté de comprender lo que me pedían además del código a corregir.*
2. *Investigué de proyectos similares en la internet además de que se trataba el problema a solucionar.*
3. *Escribí el código y compile..*
4. *Probe y corregí los errores que hubiera*
5. *Le di un buen estilo para cuando imprimiera.*

## REFERENCIAS Y BIBLIOGRAFÍA

<https://www.youtube.com/watch?v=Q1cP-yugi5M>

[https://github.com/programmerBrayan/AUCCACUSI\\_BRAYAN\\_LABORATORIO\\_02.git](https://github.com/programmerBrayan/AUCCACUSI_BRAYAN_LABORATORIO_02.git)