

# LABORATORIO 07

## ARRAYLIST BIDIMENSIONAL

**APELLIDOS Y NOMBRES:** CHAMBILLA PERCA, Valentina Milagros

**CUI:** 20213032

**CURSO:** Fundamentos de la Programación 2

### **CÓDIGO PRINCIPAL: VideoJuego3**

```
public class VideoJuego3 {  
    public static void main(String[] args) {  
        int cantidad1, cantidad2;  
  
        //Inference diamond  
  
        ArrayList<ArrayList<Soldado>> ListaSol = new ArrayList<>();  
  
        for (int i=0; i<10;i++){  
            ListaSol.add(new ArrayList<>());  
  
            for (int j = 0; j < 10; j++)  
                ListaSol.get(i).add(new Soldado());  
        }  
  
        //Inference diamond  
  
        ListaSol.add( new ArrayList<>());  
  
        //Inicia el juego  
  
        System.out.println("*****JUEGO V.  
3*****");  
  
        cantidad1 = Aleatorio(10, 1);  
        cantidad2 = Aleatorio(10, 1);  
  
        System.out.println("El primer equipo tendrá: " + cantidad1);  
        System.out.println("El segundo equipo tendrá: " + cantidad2);  
  
        /*  
        System.out.println(cantidad + " SOLDADOS.");  
        if (cantidad>5)  
            System.out.println("QUE SUERTE OuO");  
        else  
            System.out.println("Bueno..., tu suerte quiza estará en otro lado UwU");  
        */  
    }  
}
```

```

System.out.println("Se estan generando los soldados ~~");

//Generación de Soldados
GenerarSoldados(ListaSol, cantidad1, 1);
GenerarSoldados(ListaSol, cantidad2, 2);

//Mostrar Matriz Bidimensional
mostrarMatrizDeSoldados(ListaSol);

//Mostrar Soldado con más vida
System.out.println("\nMOSTRAR SOLDADO CON MAYOR CANTIDAD DE VIDA");
mostrarSoldadoMayorVida(ListaSol, 1);
mostrarSoldadoMayorVida(ListaSol, 2);

//Promedio de los puntos de Vida
System.out.println("\nMOSTRAR PROMEDIO Y NIVEL DE VIDA");
mostrarPromedioYNivelVida(ListaSol, cantidad1, 1);
mostrarPromedioYNivelVida(ListaSol, cantidad2, 2);
ArrayList<Soldado> SoldadoOrd1 = new ArrayList<Soldado>();
for (int j = 0; j < cantidad1; j++)
    SoldadoOrd1.add(new Soldado());
ArrayList<Soldado> SoldadoOrd2 = new ArrayList<Soldado>();
for (int j = 0; j < cantidad2; j++)
    SoldadoOrd2.add(new Soldado());
llenarUnidimensional(SoldadoOrd1, ListaSol,1);
llenarUnidimensional(SoldadoOrd2, ListaSol,2);

//mostrar en Orden de creación
System.out.println("\nSOLDADOS POR ORDEN DE CREACIÓN");
mostrarSoldadosOrd(SoldadoOrd1, 1);
mostrarSoldadosOrd(SoldadoOrd2, 2);

System.out.println("\nORDENAMIENTO POR BURBUJA DEL PRIMER GRUPO");
ordenarPorPuntosBurbuja(SoldadoOrd1);
mostrarSoldadosOrd(SoldadoOrd1, 1);
ordenarPorPuntosSeleccion(SoldadoOrd2);
mostrarSoldadosOrd(SoldadoOrd2, 2);
}

public static int Aleatorio(int Mayor, int Menor){

```

```
Random aleatorio = new Random();

int num = aleatorio.nextInt(Mayor-Menor+1)+Menor;

return num;

}

public static void GenerarSoldados(ArrayList<ArrayList<Soldado>> ListaSol, int cantidad, int equi){

    String nombre="";

    int vida=0, fila=0, columna=0;

    //Generar

    for (int i=0; i<cantidad; i++){

        nombre = "Soldado"+equi+"X"+i;

        vida = Aleatorio(5,1);

        while(true){

            fila = Aleatorio(9,0);

            columna = Aleatorio(9,0);

            if(ListaSol.get(fila).get(columna).getNom().equalsIgnoreCase("")){

                ListaSol.get(fila).get(columna).setSoldado(nombre, vida, fila, columna, equi);

                break;

            }

        }

    }

}

}
```

```

System.out.printf("\n%1s%6s", "\033[30m", " " + (i+1) + " ");
for (int j = 0; j < ListaSol.get(i).size(); j++){
    System.out.printf("%1s", "|");
    switch (ListaSol.get(i).get(j).getEqui()) {
        case 1:
            System.out.printf("%1s%11s", "\033[34m", ListaSol.get(i).get(j).getNom());
            break;
        case 2:
            System.out.printf("%1s%11s", "\033[35m", ListaSol.get(i).get(j).getNom());
            break;
        default:
            System.out.printf("%1s%11s", "\033[30m", ListaSol.get(i).get(j).getNom());
            break;
    }
}
System.out.println();
}
}

```

```

public static void mostrarSoldadoMayorVida(ArrayList<ArrayList<Soldado>> ListaSol, int equi) {
    System.out.println("\nEQUIPO " + equi);
    int mayor = 0;
    for (int i=0; i<ListaSol.size(); i++){
        for (int j=0; j<ListaSol.get(i).size(); j++){
            if (mayor<ListaSol.get(i).get(j).getPuntoVida() && ListaSol.get(i).get(j).getEqui()==equi)
                mayor = ListaSol.get(i).get(j).getPuntoVida();
        }
    }
    //MOSTRAR LOS SOLDADOS CON LA MAYOR CANTIDAD DE VIDA
    System.out.printf( "%10s%8s%10s%15s%8s\n", "Nombre", "Fila", "Columna", "Puntos Vida", "Equipo");
    for (int i=0; i<ListaSol.size(); i++){
        for (int j=0; j<ListaSol.get(i).size(); j++){
            if (mayor==ListaSol.get(i).get(j).getPuntoVida() && ListaSol.get(i).get(j).getEqui()==equi)

```

```

        System.out.printf( "%10s%8s%10s%15d%8d\n", ListaSol.get(i).get(j).getNom(),
ListaSol.get(i).get(j).getFila(), ListaSol.get(i).get(j).getColumna(),ListaSol.get(i).get(j).getPuntoVida(),
ListaSol.get(i).get(j).getEqui());
    }
}
}

```

```

public static void mostrarPromedioYNivelVida(ArrayList<ArrayList<Soldado>> ListaSol, int num, int equi) {
    System.out.println("\nEQUIPO " + equi);
    double suma=0;
    for (int i=0; i<ListaSol.size(); i++){
        for (int j=0; j<ListaSol.get(i).size(); j++){
            if (ListaSol.get(i).get(j).getEqui()==equi)
                suma = suma + ListaSol.get(i).get(j).getPuntoVida();
        }
    }
    double promedio = suma/num;
    //MOSTRAR LOS SOLDADOS CON LA MAYOR CANTIDAD DE VIDA
    System.out.println( "EL NIVEL DE VIDA ES: " + suma);
    System.out.println( "EL PROMEDIO DE VIDA ES: " + promedio);
}

```

```

public static void llenarUnidimensional(ArrayList<Soldado> Orden,ArrayList<ArrayList<Soldado>> ListaSol, int equi)
{
    int indice;
    String name, ulti;
    for (int i=0; i<ListaSol.size(); i++){
        for (int j=0; j<ListaSol.get(i).size(); j++){
            if (!(ListaSol.get(i).get(j).getNom().equalsIgnoreCase("")) && ListaSol.get(i).get(j).getEqui()==equi){
                name = ListaSol.get(i).get(j).getNom();
                ulti = name.substring(name.length()-1);
                indice = Integer.parseInt(ulti);
                Orden.set(indice, ListaSol.get(i).get(j));
            }
        }
    }
}

```

```
}  
}
```

```
public static void mostrarSoldadosOrd(ArrayList<Soldado> ListaSol, int equi) {  
    System.out.println("\nEQUIPO " + equi);  
    System.out.printf( "%10s%8s%10s%15s%8s\n", "Nombre", "Fila", "Columna", "Puntos Vida", "Equipo");  
    for (int i=0; i<ListaSol.size(); i++){  
        System.out.printf( "%10s%8s%10s%15d%8d\n", ListaSol.get(i).getNom(), ListaSol.get(i).getFila(),  
ListaSol.get(i).getColumna(),ListaSol.get(i).getPuntoVida(), ListaSol.get(i).getEqui());  
    }  
}
```

```
public static void ordenarPorPuntosBurbuja(ArrayList<Soldado> ListaSol){  
    System.out.println("ORDENAR POR PUNTOS: Burbuja");  
    for (int i=1; i<ListaSol.size();i++){  
        for(int j=0;j<ListaSol.size()-i;j++){  
            if((ListaSol.get(j).getPuntoVida())<(ListaSol.get(j+1).getPuntoVida()))  
                intercambiar(ListaSol, j, j+1);  
        }  
    }
```

```
private static void intercambiar(ArrayList<Soldado> ListaSol, int i, int j){  
    Soldado temp = new Soldado();  
    temp = ListaSol.get(i);  
    ListaSol.set(i,ListaSol.get(j));  
    ListaSol.set(j, temp);  
}
```

```
public static void ordenarPorPuntosSeleccion(ArrayList<Soldado> ListaSol){  
    System.out.println("ORDENAR POR PUNTOS: Selección");  
    //Encontrar el menor  
    Soldado mayor = new Soldado();  
    int indice=0;  
    for (int i=0; i<ListaSol.size();i++){  
        //Determinar el menor
```

```

    mayor = ListaSol.get(i);
    for (int j=i; j<ListaSol.size();j++)
        if (mayor.getPuntoVida() <= ListaSol.get(j).getPuntoVida()){
            mayor = ListaSol.get(j);
            indice = j;
        }
    intercambiar(ListaSol, indice, i);
}
}
}

```

### **CLASE SOLDADO: Soldado**

```

public class Soldado {
    private String Nom="";
    private int puntoVida=0;
    private int Fila=0;
    private int Columna=0;
    private int Equipo=0;
    //Equipo 0: No pertenece por lo tanto no contiene nada
    //Equipo 1: Primer equipo
    //Equipo 2: Segundo Equipo

    public void setSoldado(String nombre, int Vida, int fila, int columna, int equi){
        Nom = nombre;
        puntoVida = Vida;
        Fila = fila;
        Columna = columna;
        Equipo = equi;
    }

    public String getNom(){
        return Nom;
    }
}

```

```

public int getPuntoVida(){
    return puntoVida;
}

```

```

public int getFila(){
    return Fila;
}

```

```

public int getColumna(){
    return Columna;
}

```

```

public int getEqui(){
    return Equipo;
}

```

```

}

```

### EJEMPLOS DE APLICACIÓN

```

run:
*****JUEGO V. 3*****
El primer equipo tendrá: 7
El segundo equipo tendrá: 1
Se estan generando los soldados ~~~
MOSTRAR MATRIZ BIDIMENSIONAL DE SOLDADOS

```

F/C	A	B	C	D	E	F	G	H	I	J
1										
2					Soldado1X1			Soldado1X2	Soldado1X6	
3										
4										
5				Soldado1X5						
6										
7		Soldado1X4			Soldado2X0					
8										
9									Soldado1X3	Soldado1X0
10										

```

MOSTRAR SOLDADO CON MAYOR CANTIDAD DE VIDA

EQUIPO 1
Nombre      Fila  Columna  Puntos Vida  Equipo
Soldado1X5   4      3         5         1
Soldado1X0   8      9         5         1

EQUIPO 2

```



Nombre	Fila	Columna	Puntos Vida	Equipo
Soldado2X0	6	4	5	2

MOSTRAR PROMEDIO Y NIVEL DE VIDA

EQUIPO 1  
 EL NIVEL DE VIDA ES: 26.0  
 EL PROMEDIO DE VIDA ES: 3.7142857142857144

EQUIPO 2  
 EL NIVEL DE VIDA ES: 5.0  
 EL PROMEDIO DE VIDA ES: 5.0

SOLDADOS POR ORDEN DE CREACIÓN

EQUIPO 1

Nombre	Fila	Columna	Puntos Vida	Equipo
Soldado1X0	8	9	5	1
Soldado1X1	1	4	3	1
Soldado1X2	1	7	3	1
Soldado1X3	8	8	2	1
Soldado1X4	6	1	4	1
Soldado1X5	4	3	5	1
Soldado1X6	1	8	4	1

EQUIPO 2

Nombre	Fila	Columna	Puntos Vida	Equipo
Soldado2X0	6	4	5	2

ORDENAMIENTO POR BURBUJA DEL PRIMER GRUPO  
 ORDENAR POR PUNTOS: Burbuja

Nombre	Fila	Columna	Puntos Vida	Equipo
Soldado1X0	8	9	5	1
Soldado1X5	4	3	5	1
Soldado1X4	6	1	4	1
Soldado1X6	1	8	4	1
Soldado1X1	1	4	3	1
Soldado1X2	1	7	3	1
Soldado1X3	8	8	2	1

ORDENAR POR PUNTOS: Selección

EQUIPO 2

Nombre	Fila	Columna	Puntos Vida	Equipo
Soldado2X0	6	4	5	2

BUILD SUCCESSFUL (total time: 0 seconds)

run:  
\*\*\*\*\*JUEGO V. 3\*\*\*\*\*

El primer equipo tendrá: 10  
El segundo equipo tendrá: 8  
Se estan generando los soldados ~~~  
MOSTRAR MATRIZ BIDIMENSIONAL DE SOLDADOS

F/C	A	B	C	D	E	F	G	H	I	J
1										
2	Soldado2X0					Soldado1X4				Soldado1X0
3										
4					Soldado2X1		Soldado2X2			
5	Soldado2X4				Soldado1X1			Soldado1X9	Soldado1X3	
6		Soldado1X8								
7		Soldado2X3								
8		Soldado2X5					Soldado1X2		Soldado1X6	
9		Soldado1X5		Soldado1X7						
10	Soldado2X6		Soldado2X7							

MOSTRAR SOLDADO CON MAYOR CANTIDAD DE VIDA

EQUIPO 1					
Nombre	Fila	Columna	Puntos	Vida	Equipo
Soldado1X8	5	1	5	5	1
Soldado1X5	8	1	5	5	1
EQUIPO 2					
Nombre	Fila	Columna	Puntos	Vida	Equipo
Soldado2X7	9	2	5	5	2

MOSTRAR PROMEDIO Y NIVEL DE VIDA

EQUIPO 1  
EL NIVEL DE VIDA ES: 32.0  
EL PROMEDIO DE VIDA ES: 3.2

EQUIPO 2  
EL NIVEL DE VIDA ES: 24.0  
EL PROMEDIO DE VIDA ES: 3.0

SOLDADOS POR ORDEN DE CREACIÓN

EQUIPO 1					
Nombre	Fila	Columna	Puntos	Vida	Equipo
Soldado1X0	1	9	1	1	1
Soldado1X1	4	4	4	1	1
Soldado1X2	7	6	4	1	1
Soldado1X3	4	8	3	1	1
Soldado1X4	1	5	4	1	1
Soldado1X5	8	1	5	1	1
Soldado1X6	7	8	1	1	1
Soldado1X7	8	3	3	1	1
Soldado1X8	5	1	5	1	1
Soldado1X9	4	7	2	1	1
EQUIPO 2					
Nombre	Fila	Columna	Puntos	Vida	Equipo
Soldado2X0	1	0	3	2	2
Soldado2X1	3	4	2	2	2
Soldado2X2	3	6	1	2	2
Soldado2X3	6	1	4	2	2
Soldado2X4	4	0	4	2	2
Soldado2X5	7	1	4	2	2

Soldado2X6	9	0	1	2
Soldado2X7	9	2	5	2

ORDENAMIENTO POR BURBUJA DEL PRIMER GRUPO  
ORDENAR POR PUNTOS: Burbuja

#### EQUIPO 1

Nombre	Fila	Columna	Puntos Vida	Equipo
Soldado1X5	8	1	5	1
Soldado1X8	5	1	5	1
Soldado1X1	4	4	4	1
Soldado1X2	7	6	4	1
Soldado1X4	1	5	4	1
Soldado1X3	4	8	3	1
Soldado1X7	8	3	3	1
Soldado1X9	4	7	2	1
Soldado1X0	1	9	1	1
Soldado1X6	7	8	1	1

ORDENAR POR PUNTOS: Selección

#### EQUIPO 2

Nombre	Fila	Columna	Puntos Vida	Equipo
Soldado2X7	9	2	5	2
Soldado2X5	7	1	4	2
Soldado2X4	4	0	4	2
Soldado2X3	6	1	4	2
Soldado2X0	1	0	3	2
Soldado2X1	3	4	2	2
Soldado2X2	3	6	1	2
Soldado2X6	9	0	1	2

BUILD SUCCESSFUL (total time: 0 seconds)