INFORME DE LABORATORIO

(formato estudiante)

INFORMACIÓN BÁSICA					
ASIGNATURA:	*FUNDAMENTOS DE PROGRAMACIÓN 2*				
TÍTULO DE LA PRÁCTICA:	*Arreglos Bidimensionales de Objetos*				
NÚMERO DE PRÁCTICA:	*5*	**AÑO LECTIVO:**	*2024*	**NRO. SEMESTRE:**	*2024-B*
FECHA DE PRESENTACIÓN	*18/10/2024*	**HORA DE PRESENTACIÓN**	*18/30/00*		
INTEGRANTE (s) *Auccacusi Conde Brayan Carlos*				**NOTA (0-20)**	
DOCENTE(s): *Ing. Lino Jose Pinto Oppe*					

RESULTADOS Y PRUEBAS

I. EJERCICIOS RESUELTOS:

Ejercicio 1:

- 1. Cree un Proyecto llamado Laboratorio5
- 2. Usted deberá crear las dos clases Soldado.java y VideoJuego2.java. Puede reutilizar lo desarrollado en Laboratorio 3 y 4.
- 3. Del Soldado nos importa el nombre, nivel 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 arreglo bidimensional de objetos.
- 5. Inicializar el tablero con n soldados aleatorios entre 1 y 10. Cada soldado tendrá un nombre autogenerado: Soldado0, Soldado1, etc., un valor de nivel de vida autogenerado aleatoriamente [1..5], la fila y columna también autogenerados aleatoriamente (verificar que no puede haber 2 soldados en el mismo cuadrado). Se debe mostrar el tablero con todos los soldados creados (usar caracteres como | _ y otros). Además, mostrar los datos del Soldado con mayor nivel de vida, el promedio de nivel de vida de todos los soldados creados, el nivel de vida de todo el ejército, los datos de todos los soldados en el orden que fueron creados y un ranking de poder de todos los soldados creados, del que tiene más nivel de vida al que tiene menos (usar al menos 2 algoritmos de ordenamiento).

Clase Soldado

```java
package L5;

public class Soldado {
    private String nombre;
    private int vida;
    private int fila;
    private char columna;
    private boolean estado = false;

    //Metodos mutadores
    public void setNombre( String n){
        nombre = n;
    }
    public void setFila(int f){
        fila = f;
    }
    public void setColumna(char c){
        columna = c;
    }
    public void setVida(int v){
        vida = v;
    }
    public void setEstado(boolean e){
        estado = e;
    }


    // Metodos accesores
    public String getNombre(){
        return nombre;
    }
    public int getFila(){
        return fila;
    }
    public char getColumna(){
        return columna;
    }
    public int getVida(){
        return vida;
    }
    public boolean getEstado(){
        return estado;
    }

    public String toString(){
        return "Nombre: " + getNombre() + "    Vida: " + getVida() +
            "    Fila " + (getFila() + 1) + "    Columna: " + getColumna();
    }
}
```

Clase principal VideoJuego2

```java
package L5;
import java.util.*;
public class VideoJuego2 {
    public static void main(String [] args){
        Random rand = new Random();
        Soldado[][] campo = new Soldado[10][10];
        int i = 1;
        boolean[][] ocupados = new boolean[10][10];
        Soldado[] ordenCreacion = new Soldado[10];
        int indice = 0;
        while(i <= 10) {
            int fila = rand.nextInt(10);
            int columna = rand.nextInt(10);
            while(ocupados[fila][columna] == true){
                fila = rand.nextInt(10);
                columna = rand.nextInt(10);
            }
            ocupados[fila][columna] = true;
            Soldado soldado = new Soldado();
            if(i == 10){
                soldado.setNombre("Soldado10");
            }
            else {
                soldado.setNombre("Soldado " + i);
            }
            soldado.setVida(rand.nextInt(5) + 1);
            soldado.setFila(fila);
            soldado.setColumna((char)('A' + columna));
            soldado.setEstado(true);
            campo[fila][columna] = soldado;
            ordenCreacion[indice++] = soldado;
            i++;
        }
        mostrarTabla(campo);
        mostrarMayorVida(campo);
        promedioVida(campo);
        nivelDeVidaEjercito(campo);
        System.out.println("\nDatos de los soldados en orden de creación:");
        mostrar(campo);
        aleatorio(campo);
        System.out.println("\nAleatorio");
        mostrar(campo);
        System.out.println("\nOrdenamiento BURBUJA");
        ordenarPorVidaBurbuja(campo);
        mostrar(campo);
        aleatorio(campo);
        System.out.println("\nAleatorio");

        mostrar(campo);
        System.out.println("\nOrdenamiento SELECCION");
        ordenarPorVidaSeleccion(campo);
        mostrar(campo);
        System.out.println("\nRanking de mayor a menor: ");
        rankingMayorMenor(campo);
        mostrar(campo);

    }


```

```java

public static void mostrarTabla(Soldado[][] camp){
    for(char i = 'A'; i < 'K'; i++)
    {
        System.out.print("       " + i + "  ");
    }

    System.out.println();
    for(int i = 0; i < camp.length; i++)
    {
        if(i == 9){
            System.out.print(i + 1);
        }
        else{
            System.out.print((i + 1) + " ");
        }
        for(int j = 0; j < camp[i].length; j++)
        {
            if (camp[i][j] != null && camp[i][j].getEstado())
                System.out.print("|" + camp[i][j].getNombre());
            else
                System.out.print("|         ");
        }
        System.out.println("|");
        System.out.println("   ------------------------------------------"+
        "-----------------------------------------------------");
    }

}


public static void mostrarMayorVida(Soldado[][] campo){
    Soldado mayor = null;
    int filMayor = 0;
    int colMayor = 0;
    for (int i = 0; i < campo.length; i++){
        for(int j = 0; j < campo[i].length; j++){
            // Verifica que la posición actual tenga un soldado y que esté activo
            if (campo[i][j] != null && campo[i][j].getEstado()){
                // Si no hay soldado registrado o el actual tiene más vida que el registrado
                // Actualiza el soldado con mayor vida
                if(mayor == null || campo[i][j].getVida() > mayor.getVida()){
                    mayor = campo[i][j];
                    filMayor = i;
                    colMayor = j;
                }
            }

        }
    }
    System.out.println("El soldado con mayor nivel de vida es: ");
    System.out.println(campo[filMayor][colMayor].toString());
}


```

```java

public static void promedioVida(Soldado[][] campo){
    int suma = 0;
    // Recorre todo el campo de soldados
    for(int i = 0; i < campo.length; i++){
        for(int j = 0; j < campo[i].length; j++){
            // Si la posición actual contiene un soldado, suma su vida al total
            if(campo[i][j] != null){
                // Acumula el valor de la vida del soldado
                suma += campo[i][j].getVida();
            }
        }
    }
    System.out.println("\nEl promedio de vida de los 10 soldados es: " + (suma / campo.length));
}

public static void nivelDeVidaEjercito(Soldado[][] campo){
    int suma = 0;
    for (int i = 0; i < campo.length; i++){
        for (int j = 0; j < campo[i].length; j++){
            if (campo[i][j] != null){
                suma += campo[i][j].getVida();
            }
        }
    }
    System.out.println("\nEl nivel de vida del ejército es: " + suma);
}


public static void mostrar(Soldado[][] campo){
    System.out.println();
    // Recorre todo el campo de soldados
    for (int i = 0; i < campo.length; i++){
        // Si hay un soldado en la posición actual, imprime su información
        for (int j = 0; j < campo[i].length; j++){
            if (campo[i][j] != null){
                // Muestra los datos del soldado
                System.out.println(campo[i][j].toString());
            }
        }
    }
}



public static void ordenarPorVidaBurbuja(Soldado[][] campo) {
    Soldado temp;
    // Recorremos todo el array bidimensional
    for (int i = 0; i < campo.length; i++) {
        for (int j = 0; j < campo[i].length; j++) {
            // Comparar el soldado actual con el resto de los soldados
            for (int k = 0; k < campo.length; k++) {
                for (int l = 0; l < campo[k].length; l++) {
                    // Comparar vida de soldados e intercambiar si están desordenados
                    if (campo[i][j] != null && campo[k][l] != null) {
                        if (campo[i][j].getVida() < campo[k][l].getVida()) {
                            temp = campo[i][j];
                            campo[i][j] = campo[k][l];
                            campo[k][l] = temp;
                        }
                    }
                }
            }
        }
    }
}

```

```java

    public static void ordenarPorVidaSeleccion(Soldado[][] campo){
        Soldado temp;
        // Recorre todas las posiciones del campo
        for (int i = 0; i < campo.length; i++){
            for (int j = 0; j < campo[i].length; j++){
                int filMenor = i;
                int colMenor = j;
                // Busca el soldado con la menor vida a partir de la posición actual
                for (int k = i; k < campo.length; k++){
                    for (int l = 0; l < campo[k].length; l++){
                        // Evita revisar posiciones ya evaluadas
                        if (k == i && l <= j) {
                            continue;
                        }
                        // Cambia las coordenadas si se encuentra un soldado con vida más baja
                        if (campo[k][l] != null && campo[filMenor][colMenor] != null){
                            if (campo[k][l].getVida() < campo[filMenor][colMenor].getVida()){
                                filMenor = k;
                                colMenor = l;
                            }
                        }
                    }
                }
                // Intercambia los soldados si se encontró uno con menor vida
                if (filMenor != i || colMenor != j){
                    temp = campo[i][j];
                    campo[i][j] = campo[filMenor][colMenor];
                    campo[filMenor][colMenor] = temp;
                }
            }
        }
    }


    public static void aleatorio(Soldado[][] campo){
        Random rand = new Random();
        Soldado temp;
        // Recorre todo el campo de soldados
        for (int i = 0; i < campo.length; i++){
            for (int j = 0; j < campo[i].length; j++){
                // Genera una fila y columna aleatoria dentro de la matriz
                int randFila = rand.nextInt(campo.length);
                int randColumna = rand.nextInt(campo[0].length);
                 // Intercambia el soldado en (i, j) con uno en una posición aleatoria (randFila, randColumna)
                temp = campo[i][j];
                campo[i][j] = campo[randFila][randColumna];
                campo[randFila][randColumna] = temp;
            }
        }
    }

    public static void rankingMayorMenor(Soldado[][] campo){
        Soldado temp;
        // Recorre todo el campo de soldados
        for (int i = 0; i < campo.length; i++){
            for (int j = 0; j < campo[i].length; j++){
                // Compara cada soldado con todos los demás en el campo
                for (int k = 0; k < campo.length; k++){
                    for (int l = 0; l < campo[k].length; l++){
                        // Verifica que ambos soldados no sean nulos
                        if (campo[i][j] != null && campo[k][l] != null){
                            // Si el soldado en (i, j) tiene más vida que el soldado en
                            //(k, l), los intercambia
                            if (campo[i][j].getVida() > campo[k][l].getVida()){
                                temp = campo[i][j];
                                campo[i][j] = campo[k][l];
                                campo[k][l] = temp;
                            }
                        }
                    }
                }
            }
        }
    }
}
```

II. PRUEBAS

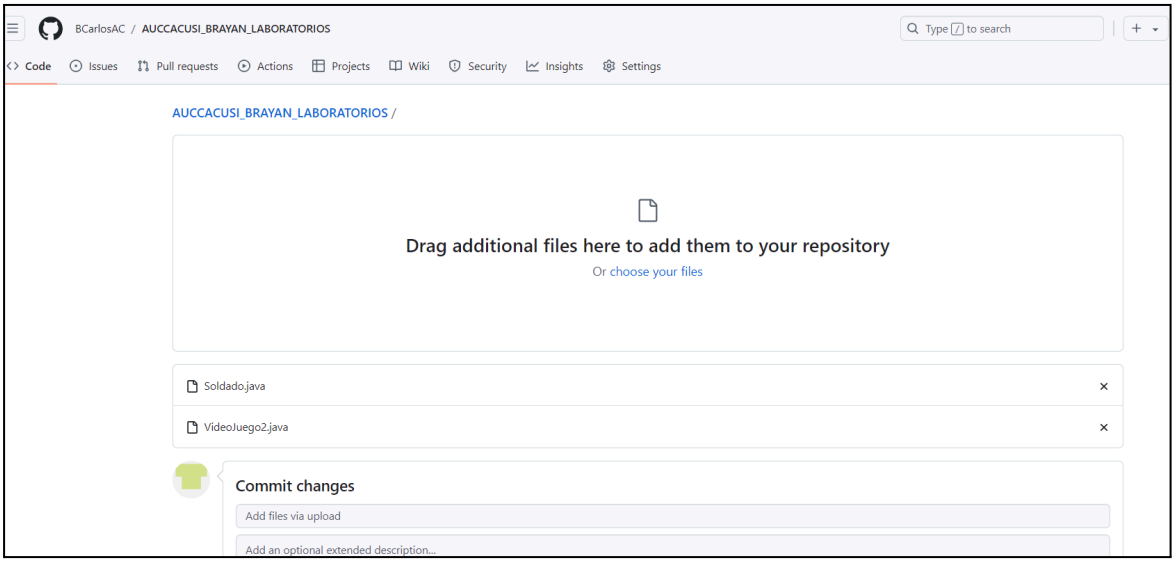
```
    A        B        C        D        E        F        G        H        I        J
1 |Soldado 5|        |        |        |        |        |        |        |Soldado10|        |
  ---------------------------------------------------------------------------------------------
2 |        |        |        |        |Soldado 4|        |        |        |        |        |
  ---------------------------------------------------------------------------------------------
3 |        |        |        |        |        |        |        |        |        |Soldado 3|
  ---------------------------------------------------------------------------------------------
4 |        |        |        |        |        |        |        |        |        |Soldado 7|
  ---------------------------------------------------------------------------------------------
5 |        |        |        |Soldado 1|        |        |        |        |        |        |
  ---------------------------------------------------------------------------------------------
6 |Soldado 6|        |        |        |        |        |        |        |        |        |
  ---------------------------------------------------------------------------------------------
7 |        |        |Soldado 8|        |        |        |        |        |        |        |
  ---------------------------------------------------------------------------------------------
8 |        |        |        |        |        |        |        |        |        |        |
  ---------------------------------------------------------------------------------------------
9 |        |        |        |        |        |        |        |        |        |        |
  ---------------------------------------------------------------------------------------------
10|        |        |Soldado 2|Soldado 9|        |        |        |        |        |        |
  ---------------------------------------------------------------------------------------------
El soldado con mayor nivel de vida es:
Nombre: Soldado 4    Vida: 5    Fila 2    Columna: E

El promedio de vida de los 10 soldados es: 2

El nivel de vida del ejército es: 26
```

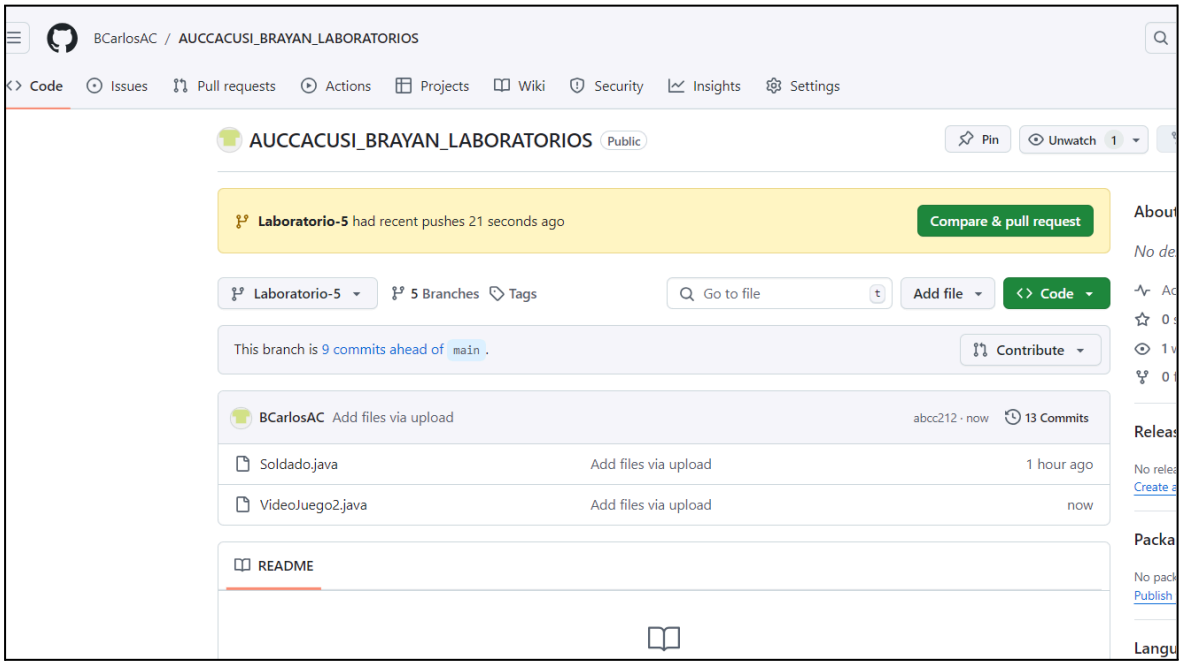
```
Datos de los soldados en orden de creación:

Nombre: Soldado 5    Vida: 2    Fila 1    Columna: A
Nombre: Soldado10    Vida: 4    Fila 1    Columna: I
Nombre: Soldado 4    Vida: 5    Fila 2    Columna: E
Nombre: Soldado 3    Vida: 2    Fila 3    Columna: J
Nombre: Soldado 7    Vida: 2    Fila 4    Columna: J
Nombre: Soldado 1    Vida: 3    Fila 5    Columna: D
Nombre: Soldado 6    Vida: 1    Fila 6    Columna: A
Nombre: Soldado 8    Vida: 4    Fila 7    Columna: C
Nombre: Soldado 2    Vida: 2    Fila 10    Columna: C
Nombre: Soldado 9    Vida: 1    Fila 10    Columna: D

Aleatorio

Nombre: Soldado 3    Vida: 2    Fila 3    Columna: J
Nombre: Soldado 2    Vida: 2    Fila 10    Columna: C
Nombre: Soldado 8    Vida: 4    Fila 7    Columna: C
Nombre: Soldado 6    Vida: 1    Fila 6    Columna: A
Nombre: Soldado 7    Vida: 2    Fila 4    Columna: J
Nombre: Soldado 5    Vida: 2    Fila 1    Columna: A
Nombre: Soldado10    Vida: 4    Fila 1    Columna: I
Nombre: Soldado 1    Vida: 3    Fila 5    Columna: D
Nombre: Soldado 9    Vida: 1    Fila 10    Columna: D
Nombre: Soldado 4    Vida: 5    Fila 2    Columna: E
```

```
Ordenamiento BURBUJA

Nombre: Soldado 6    Vida: 1    Fila 6    Columna: A
Nombre: Soldado 9    Vida: 1    Fila 10    Columna: D
Nombre: Soldado 2    Vida: 2    Fila 10    Columna: C
Nombre: Soldado 7    Vida: 2    Fila 4    Columna: J
Nombre: Soldado 5    Vida: 2    Fila 1    Columna: A
Nombre: Soldado 3    Vida: 2    Fila 3    Columna: J
Nombre: Soldado 1    Vida: 3    Fila 5    Columna: D
Nombre: Soldado10    Vida: 4    Fila 1    Columna: I
Nombre: Soldado 8    Vida: 4    Fila 7    Columna: C
Nombre: Soldado 4    Vida: 5    Fila 2    Columna: E

Aleatorio

Nombre: Soldado 9    Vida: 1    Fila 10    Columna: D
Nombre: Soldado 3    Vida: 2    Fila 3    Columna: J
Nombre: Soldado 5    Vida: 2    Fila 1    Columna: A
Nombre: Soldado 6    Vida: 1    Fila 6    Columna: A
Nombre: Soldado 1    Vida: 3    Fila 5    Columna: D
Nombre: Soldado 4    Vida: 5    Fila 2    Columna: E
Nombre: Soldado10    Vida: 4    Fila 1    Columna: I
Nombre: Soldado 2    Vida: 2    Fila 10    Columna: C
Nombre: Soldado 8    Vida: 4    Fila 7    Columna: C
Nombre: Soldado 7    Vida: 2    Fila 4    Columna: J
```

```
Ordenamiento SELECCION

Nombre: Soldado 9    Vida: 1    Fila 10    Columna: D
Nombre: Soldado 6    Vida: 1    Fila 6    Columna: A
Nombre: Soldado 5    Vida: 2    Fila 1    Columna: A
Nombre: Soldado 3    Vida: 2    Fila 3    Columna: J
Nombre: Soldado 2    Vida: 2    Fila 10    Columna: C
Nombre: Soldado 7    Vida: 2    Fila 4    Columna: J
Nombre: Soldado 1    Vida: 3    Fila 5    Columna: D
Nombre: Soldado10    Vida: 4    Fila 1    Columna: I
Nombre: Soldado 8    Vida: 4    Fila 7    Columna: C
Nombre: Soldado 4    Vida: 5    Fila 2    Columna: E

Ranking de mayor a menor:

Nombre: Soldado 4    Vida: 5    Fila 2    Columna: E
Nombre: Soldado 8    Vida: 4    Fila 7    Columna: C
Nombre: Soldado10    Vida: 4    Fila 1    Columna: I
Nombre: Soldado 1    Vida: 3    Fila 5    Columna: D
Nombre: Soldado 5    Vida: 2    Fila 1    Columna: A
Nombre: Soldado 3    Vida: 2    Fila 3    Columna: J
Nombre: Soldado 2    Vida: 2    Fila 10    Columna: C
Nombre: Soldado 7    Vida: 2    Fila 4    Columna: J
Nombre: Soldado 9    Vida: 1    Fila 10    Columna: D
Nombre: Soldado 6    Vida: 1    Fila 6    Columna: A
PS C:\Users\Hogar\Documents\BRAYAN\EP2 - Laboratories>
```

Repositorio





Link a repositorio:
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	3	
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	3	
TOTAL		20	7	18	

CONCLUSIONES

Los arreglos bidimensionales son herramientas súper útiles en programación. Me permiten organizar datos en forma de tabla, lo que facilita el acceso y la manipulación de conjuntos de datos que están relacionados. Esto se vuelve especialmente ventajoso cuando trabajo con objetos, ya que puedo representar cosas más complejas de manera clara y visual.

METODOLOGÍA DE TRABAJO

1. Revisé cuidadosamente lo que me pedían
2. Copié y retome el código del anterior trabajo
3. Comerse la parte principal y luego en función de eso desarrolle los métodos
4.Comprobé la solución, corregí los errores y modifique el estilo de impresión.

REFERENCIAS Y BIBLIOGRAFÍA