Informe de Laboratorio 05

Tema: Laboratorio 05

Nota

Estudiante	Escuela	Asignatura
Victor Mamani Anahua vmamanian@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Fundamentos de la Programación II Semestre: II Código: 20230489

Laboratorio	Tema	Duración
05	Laboratorio 05	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 09 Octubre 2023	Al 16 Octubre 2023

1. Tarea

- Cree un Proyecto llamado Laboratorio5
- Usted deberá crear las dos clases Soldado.java y VideoJuego2.java. Puede reutilizar lo desarrollado en Laboratorio 3 y 4.
- Del Soldado nos importa el nombre, puntos de vida, fila y columna (posición en el tablero).
- El juego se desarrollará en el mismo tablero de los laboratorios anteriores. Pero ahora el tablero debe ser un arreglo bidimensional de objetos.
- Inicializar el tablero con n soldados aleatorios entre 1 y 10. Cada soldado tendrá un nombre autogenerado: Soldado0, Soldado1, etc., un valor de puntos de vida autogeneradoaleatoriamente [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 (usar caracteres y () otros).
- Además de los datos del Soldado con mayor vida el promedio de puntos 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).

2. Equipos, materiales y temas utilizados

- Sistema Operativo Ubuntu GNU Linux 23 lunar 64 bits Kernell 6.2.v
- VIM 9.0.
- OpenJDK 64-Bits 19.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.
- Programación Orientada a Objetos.
- Actividades del Laboratorio 05.

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- `https://github.com/VictorMA18/fp2-23b.git`
- URL para el laboratorio 01 en el Repositorio GitHub.
- `https://github.com/VictorMA18/fp2-23b/tree/main/Fase02/Lab05`

4. Actividades del Laboratorio 05

4.1. Ejercicio Soldado

- En el primer commit explicamos el uso que le vamos a dar a esta clase para el siguiente ejercicio para esto necesitamos mas atributos como row y column para su posterior ubicacion en el tablero
- El codigo y el commit seria el siguiente:

Listing 1: Commit

```
$ git commit -m "Agregando la clase Soldado la cual reutilizamos donde agregamos un
    metodo constructor y tambien anadimos los atributos row y column con sus
    respectivos getters y setters"
```

Listing 2: Las lineas de codigos del metodo creado:

```
public class Soldado { //CREAMOS LA CLASE SOLDODADO PARA PODER USAR UN ARREGLO
    BIDIMENSIONAL DONDE NECESITAMOS LA VIDA , EL NOMBRE DEL SOLDADO Y TAMBIEN SU
    POSICION COMO LA FILA Y LA COLUMNA

  private String name;
  private int heatlh;
  private int row;
  private String column;

  //Constructor
  public Soldado(String name, int health, int row, String column){
  this.name = name;
  this.health = health;
```

```
        this.row = row;
        this.column = column;
        }

        // Metodos mutadores
        public void setName(String n){
           name = n;
        }
        public void setHealth(int p){
           heatlh = p;
        }
        public void setRow(int b){
           row = b;
        }
        public void setColumn(String c){
           column = c;
        }

        // Metodos accesores
        public String getName(){
           return name;
        }
        public int getHealth(){
           return heatlh;

        }
        public int getRow(){
           return row;
        }
        public String getColumn(){
           return column;
        }

        // Completar con otros metodos necesarios
        public String toString(){ //CREAMOS ESTE METODO PARA IMPRIMIR LOS DATOS DEl OBJETO
           String join = "\nNombre: " + getName() + "\nVida: " + getHealth() + "\nFila: " +
               getRow() + "\nColumna: " + getColumn(); //Agregamos un espaciador para poder
               separar
           return join;
        }
    }
```

4.2. Ejercicio Soldado

- En el segundo commit ponemos el nombre adecuado para el atributo salud el cual seria health y cambiamos todas sus concurrencias
- El codigo y el commit seria el siguiente:

Listing 3: Commit

```
$ git commit -m "Arreglando el nombre de la variable health en la clase soldado "
```

Listing 4: Las lineas de codigos del metodo creado:

```
    private int health;
```

4.3. Ejercicio VideoJuego2

- En el tercer commit creamos el metodo viewboard el cual nos permite dar con el tablero de manera grafica y si habria un soldado dentro de una de estas casillas se marcara con una X y si no dejara vacio
- El codigo , el commit y la ejecucion seria el siguiente:

Listing 5: Commit

```
$ git commit -m "Probando Metodo creado para poder ver el tablero para los soldados el
    cual si habria un soldado este se marcara con una X"
```

Listing 6: Las lineas de codigos del metodo creado:

```
// Laboratorio Nro 5 - Ejercicio Videojuego2
// Autor: Mamani Anahua Victor Narciso
// Colaboro:
// Tiempo:
import java.util.*;
class VideoJuego2 {
   public static void viewboard(Soldado[][] army){
      System.out.println("Mostrando tabla de posicion ... --");
      System.out.println("_____________________________________________________");
      for(int i = 0; i < army.length; i++ ){
            for(int j = 0; j < army[i].length; j++){
                  if(army[i][j].getHealth() == 0){
                        System.out.print("| " + "X" + " ");
                  }else{
                        System.out.print("|\t");
                  }
            }
            System.out.println("");
            System.out.println("|_______|_______|_______|_______|_______|_______|_______|_______|_______|___
      }
   }
   public static void main (String args[]){
      Random rdm = new Random();
      System.out.println("Cuantos soldados? ");
      int numsoldiers = rdm.nextInt(10) + 1;
      Soldado[][] army = new Soldado[10][10];
      viewboard(army);
   }
}
```

Listing 7: Ejecucion:

```
Mostrando tabla de posicion ... --
-----------------------------------------------------
Exception in thread "main" java.lang.NullPointerException: Cannot invoke
    "Soldado.getHealth()" because "<parameter1>[<local1>][<local2>]" is null
```

```
    at VideoJuego2.viewboard(VideoJuego2.java:12)
    at VideoJuego2.main(VideoJuego2.java:27)
```

4.4. Ejercicio Soldado

- En el cuarto commit creamos este metodo que nos puede ayudar a identificar casillas nulas para nuestro ejercicio con Videojuego2.java y asi poder imprimirlas
- El codigo y el commit seria el siguiente:

Listing 8: Commit

```
$ git commit -m "Arreglando el nombre de la variable health en la clase soldado "
```

Listing 9: Las lineas de codigos del metodo creado:

```
//Anadiendo metodo que nos permita que un arreglo tenga datos nulos si este esta vacio
public Soldado(){
   this.name = "";
   this.health = 0;
   this.row = 0;
   this.column = "";
}
```

4.5. Ejercicio VideoJuego2

- En el quinto commit modificamos el metodo viewboard() el cual seria de lo que imprime y modificamos la sentencia del if para que cuando un soldado de las casillas no esta vacia el cual dice que tiene datos que verifican esto entonces este retornara un X pero si esta vacia este no imprimira nada y lo dejara vacio la casilla
- El codigo , el commit y la ejecucion seria el siguiente:

Listing 10: Commit

```
$ git commit -m "Arreglando cosas para que se de la grafica del tableroen esto esta
    cambiar la condicion para que cuando esta sea null 65265206"
```

Listing 11: Las lineas de codigos del metodo creado:

```
public static void viewboard(Soldado[][] army){
   System.out.println("Mostrando tabla de posicion ... --");
   System.out.println("_____________________________________________________________________________");
   for(int i = 0; i < army.length; i++ ){
        for(int j = 0; j < army[i].length; j++){
            if(army[i][j] != null){
                System.out.print("| " + "X" + " ");
            }else{
                System.out.print("|\t");
            }
        }
        System.out.println("|");
```

```
        System.out.println("|_______|_______|_______|_______|_______|_______|_______|_______|_______|_____
    }
}
```

Listing 12: Las lineas de codigos del metodo creado: VER EL TEXTO EN LATEX EN LA IMAGEN SE DEFORMA O EJECUTARLO

```
Cuantos soldados?
Mostrando tabla de posicion ... --
--------------------------------------------------------------------------------
|       |       |       |       |       |       |       |       |       |       |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|       |       |       |       |       |       |       |       |       |       |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|       |       |       |       |       |       |       |       |       |       |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|       |       |       |       |       |       |       |       |       |       |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|       |       |       |       |       |       |       |       |       |       |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|       |       |       |       |       |       |       |       |       |       |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|       |       |       |       |       |       |       |       |       |       |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|       |       |       |       |       |       |       |       |       |       |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|       |       |       |       |       |       |       |       |       |       |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|       |       |       |       |       |       |       |       |       |       |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
```

4.6. Ejercicio VideoJuego2

- En el SEXTO commit creamos el metodo fillarray() el cual queremos rellenar el array de soldados para poder verlos en el tablero a la ves su informacion en este commit se prodro un resultado el cual seria de la columna y el resultado de esta con una imprimicion ya que no estaria seguro de su resultado
- El codigo , el commit y la ejecucion seria el siguiente:

Listing 13: Commit

```
$ git commit -m "Anadiendo el metodo fillarray() el cual va a llenar nuestro arreglo de
    soldados para eso necesitabamos su nombre su vida su fila y su columna la cual yo
    quiero probar la cilumna lo que imprime para que compruebe que este saliendo lo
    esperado un letra la cual nos de su posicion"
```

Listing 14: Las lineas de codigos del metodo creado:

```
public static Soldado[][] fillarray(int number){
   Random rdm = new Random();
   Soldado[][] army= new Soldado[10][10];
   for(int i = 0; i < number; i++){
         String name = "Soldado" + (i + 1);
```

```
            int health = rdm.nextInt(5) + 1;
            int row = rdm.nextInt(10) + 1;
            String column = String.valueOf((char)(rdm.nextInt(10) + 65));
            System.out.println(column);
        }
        return null;
    }
```

Listing 15: Las lineas de codigos del metodo creado: VER EL TEXTO EN LATEX EN LA IMAGEN SE DEFORMA O EJECUTARLO

```
Cuantos soldados?
1
Mostrando tabla de posicion ... --
---------------------------------------------------------------------------------
|      |      |      |      |      |      |      |      |      |      |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|      |      |      |      |      |      |      |      |      |      |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|      |      |      |      |      |      |      |      |      |      |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|      |      |      |      |      |      |      |      |      |      |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|      |      |      |      |      |      |      |      |      |      |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|      |      |      |      |      |      |      |      |      |      |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|      |      |      |      |      |      |      |      |      |      |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|      |      |      |      |      |      |      |      |      |      |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|      |      |      |      |      |      |      |      |      |      |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
|      |      |      |      |      |      |      |      |      |      |
|_______|_______|_______|_______|_______|_______|_______|_______|_______|_______|
F
```

4.7. Ejercicio VideoJuego2

- En el septimo commit completamos el metodo fillarray() el cual ya usamos el constructor soldado ya que tendriamos los datos despues de esto comprobariamos si este casillero usado seria algo vacio y se podria llenarlo y si no este retrocederia una iteracion del ciclo ya que se estaria usando uno para preguntar a la ves tambien imprimimos los datos del soldado creado
- El codigo , el commit y la ejecucion seria el siguiente:

Listing 16: Commit

```
$ git commit -m "Completando el metodo fillarray() para que este pueda imprmir los
    datos de los soldados que estan en el tablero este te devuelve el array lleno con
    la cantidad de soldados y texto que dice su informacion de cada soldado Y tambien
    cumplimos con el cual no se repita un soldado en el mismo casillero ya que al ver
    que este lleno este retrocedera una repeticion ya que estaria tomando la repeticion
    en un casillero lleno"
```

Listing 17: Las lineas de codigos del metodo creado:

```
public static Soldado[][] fillarray(int number){
   Random rdm = new Random();
   Soldado[][] army= new Soldado[10][10];
   System.out.println("Registrando soldados .......");
   for(int i = 0; i < number; i++){
         String name = "Soldado" + (i); //CORREGIMOS EN EL NOMBRE YA QUE ESTE COMNEZABA
             DESDE EL 0
         int health = rdm.nextInt(5) + 1;
         int row = rdm.nextInt(10) + 1;
         String column = String.valueOf((char)(rdm.nextInt(10) + 65));
         if(army[row - 1][(int)column.charAt(0) - 65] == null){
              System.out.print("**********************************");
              army[row - 1][(int)column.charAt(0) - 65] = new Soldado(name, health, row,
                  column);
              System.out.println(army[row - 1][(int)column.charAt(0) - 65].toString());
         }else{
              i -= 1;
         }
   }
   return army;
}
```

Listing 18: Las lineas de codigos del metodo creado: VER EL TEXTO EN LATEX EN LA IMAGEN SE DEFORMA O EJECUTARLO

```
Cuantos soldados?
8
Registrando soldados .......
**********************************
Nombre: Soldado1
Vida: 2
Fila: 10
Columna: D
**********************************
Nombre: Soldado2
Vida: 2
Fila: 6
Columna: E
**********************************
Nombre: Soldado3
Vida: 5
Fila: 5
Columna: F
**********************************
Nombre: Soldado4
Vida: 1
Fila: 9
Columna: I
**********************************
Nombre: Soldado5
Vida: 2
Fila: 9
Columna: E
**********************************
```

```
Nombre: Soldado6
Vida: 4
Fila: 2
Columna: F
********************************
Nombre: Soldado7
Vida: 1
Fila: 4
Columna: H
********************************
Nombre: Soldado8
Vida: 4
Fila: 3
Columna: E
Mostrando tabla de posicion ... --
```

					X				
				X					
						X			
					X				
				X					
				X			X		
			X						

4.8. Ejercicio VideoJuego2

- En el octavo commit creamos el metodo longerLife() el cual este nos permite a dar la informacion del soldado con mayor vida del ejercito el cual comparamos con otros soldados que esten en este ejercito y dependiendo de eso va a recoger informacion del soldado con mayor vida y lo guarda en soldier el cual sera imprimido despues.
- El codigo , el commit y la ejecucion seria el siguiente:

Listing 19: Commit

```
$ git commit -m "Creamos el metodo longerLife() el cual nos imprimira el dato del
    soldado con mayor vida dependiendo de los que esten en el ejercito que va
    comparando con los demas para ver quien es el mayor de todos"
```

Listing 20: Las lineas de codigos del metodo creado:

```
public static void longerLife(Soldado[][] army){
   int mayor = 0;
   Soldado soldier = null;
   for(int i = 0; i < army.length; i++){
        for(int j = 0; j < army[i].length; j++){
             if(army[i][j] != null){
                  if(mayor < army[i][j].getHealth()){
                       mayor = army[i][j].getHealth();
                       soldier = army[i][j];
                  }
             }
        }
   }
   System.out.println("");
   System.out.print("El primer soldado con mayor vida es: ");
   System.out.println(soldier.toString());
   System.out.println("********************************");
   }
```

Listing 21: Las lineas de codigos del metodo creado: VER EL TEXTO EN LATEX EN LA IMAGEN SE DEFORMA O EJECUTARLO

```
Cuantos soldados?
6
Registrando soldados .......
********************************
Nombre: Soldado0
Vida: 1
Fila: 9
Columna: C
********************************
Nombre: Soldado1
Vida: 3
Fila: 10
Columna: E
********************************
Nombre: Soldado2
Vida: 2
Fila: 4
Columna: I
********************************
Nombre: Soldado3
Vida: 3
Fila: 7
Columna: F
********************************
Nombre: Soldado4
Vida: 3
Fila: 8
Columna: H
********************************
Nombre: Soldado5
Vida: 2
Fila: 1
Columna: G
Mostrando tabla de posicion ... --
```

```
_____________________________________________________________________________
|      |      |      |      |      |      |   X   |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|______|
|      |      |      |      |      |      |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|______|
|      |      |      |      |      |      |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|______|
|      |      |      |      |      |      |      |      |   X  |      |      |
|______|______|______|______|______|______|______|______|______|______|______|
|      |      |      |      |      |      |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|______|
|      |      |      |      |      |      |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|______|
|      |      |      |      |      |   X  |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|______|
|      |      |      |      |      |      |      |   X  |      |      |      |
|______|______|______|______|______|______|______|______|______|______|______|
|      |      |   X  |      |      |      |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|______|
|      |      |      |      |   X  |      |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|______|

El primer soldado con mayor vida es:
Nombre: Soldado3
Vida: 3
Fila: 7
Columna: F
*********************************
```

4.9. Ejercicio VideoJuego2

- En el noveno commit creamos el metodo averageLife() el cual nos retorna la imprimicion del promedio de vida de todos los soldados del ejercito el cual se va recolectando para despues dividirlo por la cantidad de soldados el cual va a ser decimal para eso necesitamos hacerlo double.
- El codigo , el commit y la ejecucion seria el siguiente:

Listing 22: Commit

```
$ git commit -m "Agregamos el metodo averageLife() el cual nos dara el promedio de vida
    del ejercito el cual recolecteria la vida de todos y despues se dividiria en la
    cantidad de soldados el cual lo multiplicamos por 1.0 para que este sea double
    decimal y sea mas especifico y despues se imprimiria este resultado"
```

Listing 23: Las lineas de codigos del metodo creado:

```
public static void averageLife(Soldado[][] army){
   int sum = 0;
   int cont = 0;
   for(int i = 0; i < army.length; i++){
        for(int j = 0; j < army.length; j++){
            if(army[i][j] != null){
                sum += army[i][j].getHealth();
```

```
                    cont++;
                }
            }
        }
        double avg = sum / (cont * 1.0);
        System.out.println("El promedio de vida del ejercito es : " + avg);
        }
```

Listing 24: Las lineas de codigos del metodo creado: VER EL TEXTO EN LATEX EN LA IMAGEN SE DEFORMA O EJECUTARLO

```
Cuantos soldados?
10
Registrando soldados .......
*********************************
Nombre: Soldado0
Vida: 3
Fila: 5
Columna: E
*********************************
Nombre: Soldado1
Vida: 3
Fila: 4
Columna: F
*********************************
Nombre: Soldado2
Vida: 4
Fila: 4
Columna: A
*********************************
Nombre: Soldado3
Vida: 3
Fila: 2
Columna: I
*********************************
Nombre: Soldado4
Vida: 4
Fila: 9
Columna: A
*********************************
Nombre: Soldado5
Vida: 1
Fila: 1
Columna: B
*********************************
Nombre: Soldado6
Vida: 1
Fila: 1
Columna: C
*********************************
Nombre: Soldado7
Vida: 4
Fila: 4
Columna: E
*********************************
Nombre: Soldado8
```

```
Vida: 2
Fila: 10
Columna: A
*********************************
Nombre: Soldado9
Vida: 4
Fila: 5
Columna: G
Mostrando tabla de posicion ... --
 _____ _____ _____ _____ _____ _____ _____ _____ _____ _____
|     |  X  |  X  |     |     |     |     |     |     |     |
|_____|_____|_____|_____|_____|_____|_____|_____|_____|_____|
|     |     |     |     |     |     |     |  X  |     |     |
|_____|_____|_____|_____|_____|_____|_____|_____|_____|_____|
|     |     |     |     |     |     |     |     |     |     |
|_____|_____|_____|_____|_____|_____|_____|_____|_____|_____|
|  X  |     |     |     |  X  |  X  |     |     |     |     |
|_____|_____|_____|_____|_____|_____|_____|_____|_____|_____|
|     |     |     |     |  X  |     |  X  |     |     |     |
|_____|_____|_____|_____|_____|_____|_____|_____|_____|_____|
|     |     |     |     |     |     |     |     |     |     |
|_____|_____|_____|_____|_____|_____|_____|_____|_____|_____|
|     |     |     |     |     |     |     |     |     |     |
|_____|_____|_____|_____|_____|_____|_____|_____|_____|_____|
|     |     |     |     |     |     |     |     |     |     |
|_____|_____|_____|_____|_____|_____|_____|_____|_____|_____|
|  X  |     |     |     |     |     |     |     |     |     |
|_____|_____|_____|_____|_____|_____|_____|_____|_____|_____|
|  X  |     |     |     |     |     |     |     |     |     |
|_____|_____|_____|_____|_____|_____|_____|_____|_____|_____|

El primer soldado con mayor vida es:
Nombre: Soldado2
Vida: 4
Fila: 4
Columna: A
*********************************
El promedio de vida del ejercito es : 2.9
```

4.10. Ejercicio VideoJuego2

- En el decimo commit creamos el metodo rankingBurbujaHealth() el cual aplicamos que en un arreglo unidimensional ponemos todos los soldados del arreglo bidimensional para asi poderaplicar efectivamente el metodo burbuja el cual aplicariamos con el atributo health el cual nos dara un ranking que despues se imprimira los resultados del ranking del mayor vida al menor que se imprimiran

- El codigo , el commit y la ejecucion seria el siguiente:

Listing 25: Commit

```
$ git commit -m "Creando el metodo rankingBurbujaHealth() el cual aplicamos que en un
    arreglo unidimensional ponemos todos los soldados del arreglo bidimensional para
    asi poderaplicar efectivamente el metodo burbuja el cual aplicariamos con el
```

```
        atributo health el cual nos dara un ranking que despues se imprimira los resultados
        del ranking del mayor vida al menor"
```

Listing 26: Las lineas de codigos del metodo creado:

```
    public static void rankingBurbujaHealth(Soldado[][] army, int numsoldiers){
       Soldado[] soldiers = new Soldado[numsoldiers];
       int count = 0;
       Soldado soldier = null;
       for(int i = 0; i < army.length; i++){ //CREAMOS ARREGLO PARA QUE LOS SOLDADOS SE
           TRASLADEN DE UN ARREGLO BIDIMENSIONAL A UNO DIMENSIONAL PARA APLICAR EL METODO
           BURBUJA
            for(int j = 0; j < army[i].length; j++){
                if(army[i][j] != null){
                    soldiers[count] = army[i][j];
                    count++;
                }
            }
       }
       System.out.println("Ordenando a los soldados por el metodo burbuja: "); //APLICAMOS
           EL METODO BURBUJA CON LOS PUNTOS DE VIDA
       for(int i = 0; i < numsoldiers - 1; i++){
            for(int j = 0; j < numsoldiers - i - 1; j++){
                if(soldiers[j].getHealth() < soldiers[j + 1].getHealth()){
                    soldier = soldiers[j];
                    soldiers[j] = soldiers[j + 1];
                    soldiers[j + 1] = soldier;
                }
            }
       }
       System.out.println("----------------------------------------");
       System.out.println("Mostrando Ranking...");
       for(int i = 0; i < soldiers.length; i++){
            System.out.print("\n" + "Puesto " + (i + 1));
            System.out.println(soldiers[i].toString());
            System.out.println("------------------");
       }
       System.out.println("********************************");
    }
```

Listing 27: Las lineas de codigos del metodo creado: VER EL TEXTO EN LATEX EN LA IMAGEN SE DEFORMA O EJECUTARLO

```
Cuantos soldados?
6
Registrando soldados .......
********************************
Nombre: Soldado0
Vida: 3
Fila: 7
Columna: I
********************************
Nombre: Soldado1
Vida: 1
Fila: 5
```

```
Columna: C
*********************************
Nombre: Soldado2
Vida: 1
Fila: 10
Columna: E
*********************************
Nombre: Soldado3
Vida: 2
Fila: 3
Columna: C
*********************************
Nombre: Soldado4
Vida: 4
Fila: 6
Columna: D
*********************************
Nombre: Soldado5
Vida: 1
Fila: 1
Columna: H
Mostrando tabla de posicion ... --
-------------------------------------------------------------------------------
|      |      |      |      |      |      |      |  X   |      |      |
|______|______|______|______|______|______|______|______|______|______|
|      |      |      |      |      |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|
|      |      |  X   |      |      |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|
|      |      |      |      |      |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|
|      |      |  X   |      |      |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|
|      |      |      |  X   |      |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|
|      |      |      |      |      |      |      |   X  |      |      |
|______|______|______|______|______|______|______|______|______|______|
|      |      |      |      |      |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|
|      |      |      |      |      |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|
|      |      |      |      |  X   |      |      |      |      |      |
|______|______|______|______|______|______|______|______|______|______|

El primer soldado con mayor vida es:
Nombre: Soldado4
Vida: 4
Fila: 6
Columna: D
*********************************
El promedio de vida del ejercito es : 2.0
*********************************
Ordenando a los soldados por el metodo burbuja:
-----------------------------------------
Mostrando Ranking...
```

```
Puesto 1
Nombre: Soldado4
Vida: 4
Fila: 6
Columna: D
------------------

Puesto 2
Nombre: Soldado0
Vida: 3
Fila: 7
Columna: I
------------------

Puesto 3
Nombre: Soldado3
Vida: 2
Fila: 3
Columna: C
------------------

Puesto 4
Nombre: Soldado5
Vida: 1
Fila: 1
Columna: H
------------------

Puesto 5
Nombre: Soldado1
Vida: 1
Fila: 5
Columna: C
------------------

Puesto 6
Nombre: Soldado2
Vida: 1
Fila: 10
Columna: E
------------------
**********************************
```

4.11. Estructura de laboratorio 05

- El contenido que se entrega en este laboratorio05 es el siguiente:

```
/Lab05     "Poner rama"
```

5. Rúbricas

5.1. Entregable Informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y facil de leer.

5.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumplio con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos lo items.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y demostración

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	1	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1	
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 estan dentro de los plazos de fecha de entrega establecidos.	2	X	0.5	
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		12.5	

6. Referencias

- https://drive.google.com/file/d/1CoQAKeKW-QDYRmHLrBdbSopFB1Z_Qmk3/view