Universidad Nacional San Agustín de Arequipa

Facultad de Ingeniería de Producción y Servicios

Escuela Profesional de Ingeniería de Sistemas

Asignatura:

Fundamentos de la Programación 2

Tema:

Laboratorio – Practica 8

Docente:

Revilla Arroyo Christian

Estudiante:

Cuno Cahuari Armando Steven

2023

Informe de Laboratorio 8

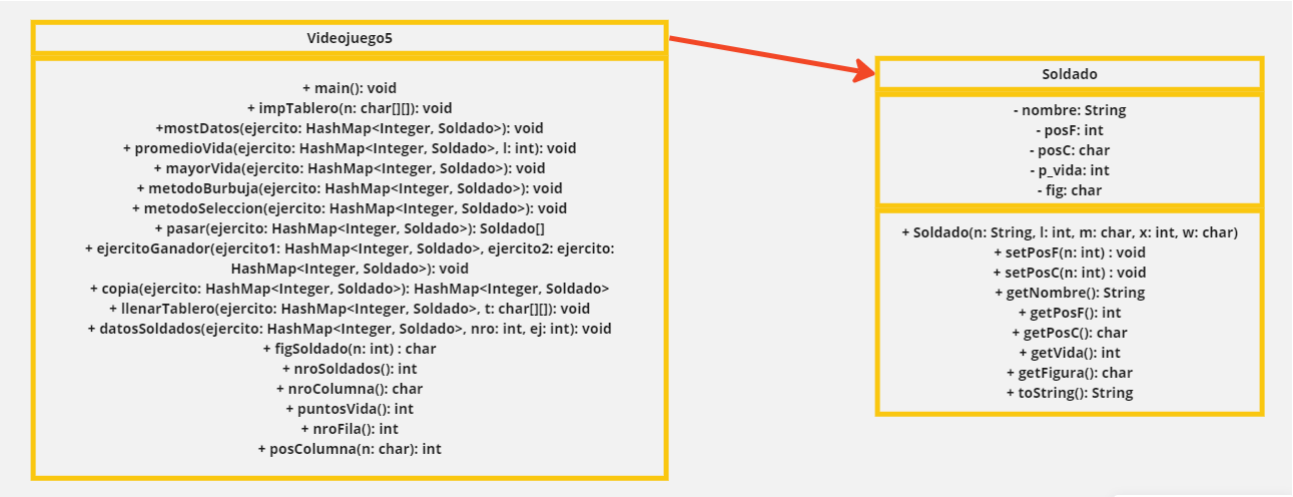
Tema: HashMap

Actividad: Laboratorio 8

URL Repositorio GitHub: `https://github.com/SteveArms/LaboratorioFPII.git`
Tiempo: 4 horas

Actividad:

Diagrama de clases UML



Clase Soldado:

Para empezar debemos crear la clase Soldado la cual tenga los tributos nombre, posición tanto en fila como columna, puntos de vida y la figura que lo representa. Para ello acompañado de sets y gets.

```java
public class Soldado {
    private String nombre;
    private int posf;
    private char posc;
    private int p_vida;
    private char fig;
    public Soldado(String n, int l, char m, int x, char w){
        nombre = n;
        posf = l;
        posc = m;
        p_vida = x;
        fig = w;
    }
    public void setPosF(int n){
        posf = n;
    }
    public void setPosC(char n){
        posc = n;
    }
    public String getNombre(){
        return nombre;
    }
    public int getPosF(){
        return posf;
    }
    public char getPosC(){
        return posc;
    }
    public int getVida(){
        return p_vida;
    }
    public char getFigura(){
        return fig;
    }
    @Override
    public String toString(){
        return "Nombre: " + nombre + " Vida: " + p_vida + " Posicion: " + posf + "," + posc + " Figura: " + fig;
    }
}
```

Clase Videojuego5:

Métodos para los datos de los Soldados:

Los siguientes métodos llevan la misma mecánica que los laboratorios 6 y 7 los cuales son encargados de darles valores aleatorios a cada soldado por HashMap

```
214     public static char figSoldado(int n){
215         if(n == 1)
216             return '*';
217         return '&';
218     }
219     public static int nroSoldados(){
220         return (int)(Math.random() * 9 + 1);
221     }
222     public static char nroColumna(){
223         String n = "abcdefghij";
224         int m = (int)(Math.random() * 9);
225         return n.charAt(m);
226     }
227     public static int puntosVida(){
228         return (int)(Math.random() * 4 + 1);
229     }
230     public static int nroFila(){
231         return (int)(Math.random() * 9 + 1);
232     }
233     public static int posColumna(char n){
234         switch(n){
235             case 'a': return 0;
236             case 'b': return 1;
237             case 'c': return 2;
238             case 'd': return 3;
239             case 'e': return 4;
240             case 'f': return 5;
241             case 'g': return 6;
242             case 'h': return 7;
243             case 'i': return 8;
244             case 'j': return 9;
245             default: return 0;
246         }
247     }
248 }
```

Una vez tenemos los datos por cada Soldado necesitaremos tanto como imprimir el tablero y actualizar las posiciones de los Soldados sin repetir la ubicación.

```java
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        char[][] tablero ={{'-','-','-','-','-','-','-','-','-','-',},
                           {'-','-','-','-','-','-','-','-','-','-',},
                           {'-','-','-','-','-','-','-','-','-','-',},
                           {'-','-','-','-','-','-','-','-','-','-',},
                           {'-','-','-','-','-','-','-','-','-','-',},
                           {'-','-','-','-','-','-','-','-','-','-',},
                           {'-','-','-','-','-','-','-','-','-','-',},
                           {'-','-','-','-','-','-','-','-','-','-',},
                           {'-','-','-','-','-','-','-','-','-','-',},
                           {'-','-','-','-','-','-','-','-','-','-',}};
        HashMap<Integer, Soldado> ejer1 = new HashMap<Integer, Soldado>();
        HashMap<Integer, Soldado> ejer2 = new HashMap<Integer, Soldado>();
        int n1 = nroSoldados();
        System.out.println("La cantidad de soldado del 1er ejertico es " + n1);
        int n2 = nroSoldados();
        System.out.println("La cantidad de soldado del 2do ejertico es " + n2);
        datosSoldados(ejer1, n1, 1);
        datosSoldados(ejer2, n2, 2);
        llenarTablero(ejer1, tablero);
        llenarTablero(ejer2, tablero);
    public static void datosSoldados(HashMap<Integer, Soldado> ejercito, int nro, int ej){
        for(int i = 0; i < nro; i++){
            String nombre = "Soldado" + (i + 1) + "X" + ej;
            Soldado p = new Soldado(nombre, nroFila(), nroColumna(), puntosVida(), figSoldado(ej));
            ejercito.put(i, p);
        }
    }
    public static void llenarTablero(HashMap<Integer, Soldado> ejercito, char[][] t){
        for(int i : ejercito.keySet()){
            Soldado n = ejercito.get(i);
            int fila = n.getPosF();
            char columna = n.getPosC();
            while(t[fila][posColumna(columna)] != '-'){
                fila = nroFila();
                columna = nroColumna();
            }
            t[fila][posColumna(columna)] = n.getFigura();
            n.setPosF(fila);
            n.setPosC(columna);
        }
    }
```

Métodos:

- La primera opción imprimirá un tablero en referencia a la posición de todos los soldados recorriendo char por char mediante un arreglo bidimensional con ayuda de ciclos for anidados

```
90      public static void impTablero(char[][] n){
91          for(int i = 0; i < n.length; i++){
92              for(int j = 0; j < n[i].length; j++){
93                  System.out.print(n[i][j]);
94              }
95              System.out.println();
96          }
97      }
```

- La segunda y tercera opción se encarga de mostrar los soldados con mayor vida de cada ejercito. Primero hace una búsqueda mediante un ciclo for guardando en una variable de tipo Int la mayor vida que vaya encontrando. De ahí realizamos algo parecido a la búsqueda lineal en la cual constara que haga un recorrido por HashMap de cada ejercito e imprimira los datos que se igualen a la vida mayor que se encontró por ejercito.

```
111     public static void mayorVida(HashMap<Integer, Soldado> ejercito){
112         int mayor = 0;
113         for(Soldado i: ejercito.values()){
114             if(i.getVida() > mayor){
115                 mayor = i.getVida();
116             }
117         }
118         for(Soldado n: ejercito.values()){
119             if(n.getVida() == mayor){
120                 System.out.println(n);
121             }
122         }
123     }
```

- La cuarta y quinta opción muestra el promedio de vida por cada ejército en el cual con apoyo de una variable tipo doublé el cual almacenara la suma de puntos de vida que vaya recorriendo mediante el ciclo, de ahí dividas por el tamaño del HashMap

```
103     public static void promedioVida(HashMap<Integer, Soldado> ejercito, int l){
104         double n = 0;
105         for(Soldado m: ejercito.values()){
106             n += m.getVida();
107         }
108         n = n / ejercito.size();
109         System.out.println("El promedio de vida del ejercito " + l + " es " + n);
110     }
```

- La sexta y séptima opción nos muestra los datos de cada ejército, acá asimismo implementamos el método de instancia creado en la clase Soldado llamado toString que se imprime de manera más asequible

```
98      public static void mostDatos(HashMap<Integer, Soldado> ejercito){
99          for(Soldado i : ejercito.values()){
100             System.out.println(i);
101         }
102     }
```

- La octava, novena, decima y onceava opción se encarga de ordenas los puntos de vida por cada ejercito de Mayor a Menor mediante 2 métodos los cuales son Burbuja y Selección para ellos hemos hecho copia del HashMap Original hacia otros 4 los cuales, para no modificar el principal.

```
25          HashMap<Integer, Soldado> copiaMetodoEjer1 = copia(ejer1);
26          HashMap<Integer, Soldado> copiaMetodo2Ejer1 = copia(ejer1);;
27          HashMap<Integer, Soldado> copiaMetodoEjer2 = copia(ejer2);
28          HashMap<Integer, Soldado> copiaMetodo2Ejer2 = copia(ejer2);
```

```
183 public static HashMap<Integer, Soldado> copia(HashMap<Integer, Soldado> ejercito){
184     HashMap<Integer, Soldado> copiaOriginal = new HashMap<Integer, Soldado>();
185     for(int i = 0; i < ejercito.size(); i++){
186         Soldado orig = ejercito.get(i);
187         Soldado nuevo = new Soldado(orig.getNombre(), orig.getPosF(), orig.getPosC(), orig.getVida(), orig.getFigura());
188         copiaOriginal.put(i,nuevo);
189     }
190     return copiaOriginal;
191 }
```

Una ves realizamos esto implementaremos de 2 maneras ambos métodos tanto Burbuja y Selección:

```
127     public static void metodoBurbuja(HashMap<Integer, Soldado> ejercito){
128         for(int i = 0; i < ejercito.size() - 1; i++){
129             for(int y = 0; y < ejercito.size() - 1; y++){
130                 Soldado m = ejercito.get(y);
131                 Soldado n = ejercito.get(y + 1);
132                 if(m.getVida() < n.getVida()){
133                     Soldado menor = new Soldado(m.getNombre(), m.getPosF(), m.getPosC(), m.getVida(), m.getFigura());
134                     Soldado mayor = new Soldado(n.getNombre(), n.getPosF(), n.getPosC(), n.getVida(), n.getFigura());
135                     ejercito.put(y, mayor);
136                     ejercito.put(y + 1, menor);
137                 }
138             }
139         }
140         mostDatos(ejercito);
141     }
```

El método burbuja de ir comparando de a 2 elementos y para ello. Se hace un cambio de valores mediante las llaves y valores que se contiene en el HashMap

```
142     public static void metodoSeleccion(HashMap<Integer, Soldado> ejercito){
143         Soldado[] nt = pasar(ejercito);
144         for(int i = 0; i < nt.length - 1 ; i++){
145             for(int j = i + 1; j < nt.length - 1; j++){
146                 if(nt[i].getVida() < nt[j].getVida()){
147                     Soldado menor = new Soldado(nt[i].getNombre(), nt[i].getPosF(), nt[i].getPosC(), nt[i].getVida(), nt[i]
148                     Soldado mayor = new Soldado(nt[j].getNombre(), nt[j].getPosF(), nt[j].getPosC(), nt[j].getVida(), nt[j]
149                     nt[i] = mayor;
150                     nt[j] = menor;
151                 }
152             }
153         }
154         ejercito.clear();
155         for(int x = 0; x < nt.length; x++){
156             ejercito.put(x, nt[x]);
157         }
158         mostDatos(ejercito);
159     }
```

Para el método de Selección utilizamos una técnica diferente en el cual consiste pasar todos los datos de un HashMap hacia un arreglo de Soldado asimismo en el mismo arreglo ordenar de mayor a menor según puntos de vida, de ahí agregarle tanto sus llaves y sus valores en forma ordenada.

```
142     public static void metodoSeleccion(HashMap<Integer, Soldado> ejercito){
143         Soldado[] nt = pasar(ejercito);
144         for(int i = 0; i < nt.length - 1 ; i++){
145             for(int j = i + 1; j < nt.length - 1; j++){
146                 if(nt[i].getVida() < nt[j].getVida()){
147                     Soldado menor = new Soldado(nt[i].getNombre(), nt[i].getPosF(), nt[i].getPosC(), nt[i].getVida(), nt[i
148                     Soldado mayor = new Soldado(nt[j].getNombre(), nt[j].getPosF(), nt[j].getPosC(), nt[j].getVida(), nt[j
149                     nt[i] = mayor;
150                     nt[j] = menor;
151                 }
152             }
153         }
154         ejercito.clear();
155         for(int x = 0; x < nt.length; x++){
156             ejercito.put(x, nt[x]);
157         }
158         mostDatos(ejercito);
159     }
160     public static Soldado[] pasar(HashMap<Integer, Soldado> ejercito){
161         Soldado[] n = new Soldado[ejercito.size()];
162         for(int i = 0; i < n.length; i++){
163             n[i] = ejercito.get(i);
164         }
165         return n;
166     }
```

Una pequeña conclusión acerca del método de ordenamiento en HashMap:

Ambos pueden ser formas más viables por las cuales podemos hacer ordenamiento de datos en HashMap, sin embargo, hay otras formas que son más viables en los cuales podemos hacer uso de HashMap en los que necesitemos de claves y valores. Gracias a los métodos que nos ofrecen podremos hacer con este búsquedas más eficientes y rápidas. Almacenamiento de valores mediante llaves, entre otros.

- La opción doceava es que contabiliza la suma de puntos por diferentes ejércitos y sale victorioso el que tiene mayor al otro.

```java
public static void ejercitoGanador(HashMap<Integer, Soldado> ejercito1, HashMap<Integer, Soldado> ejercito2){
    int ejer1 = 0, ejer2 = 0;
    for(Soldado m: ejercito1.values()){
        ejer1 += m.getVida();
    }
    for(Soldado n: ejercito2.values()){
        ejer2 += n.getVida();
    }
    if(ejer1 == ejer2){
        System.out.println("Existe un empate");
    } else if(ejer1 > ejer2){
        System.out.println("El ejercito 1 es el ganador con un total de " + ejer1 + " puntos");
    } else {
        System.out.println("El ejercito 2 es el ganador con un total de " + ejer2+ " puntos");
    }
}
```

Prueba:

```
La cantidad de soldado del 1er ejertico es 9
La cantidad de soldado del 2do ejertico es 8
Opcion 1: Imprimir Tablero
Opcion 2: Soldado con mayor vida del ejercito1
Opcion 3: Soldado con mayor vida del ejercito2
Opcion 4: Promedio de vida del ejercito 1
Opcion 5: Promedio de vida del ejercito 2
Opcion 6: Mostrar datos del ejercito1
Opcion 7: Mostrar datos del ejercito2
Opcion 8: 1er metodo de orden Ejercito 1
Opcion 9: 2do metodo de orden Ejercito 1
Opcion 10: 1er metodo de orden Ejercito 1
Opcion 11: 2do metodo de orden Ejercito 2
Opcion 12: Ejercito ganador
Opcion 13: Salir
```

```
1
El tablero es :
----------
--&-------
------&---
&&-**&----
--------&-
*-&-----*-
----------
--------*-
**------*-
----*-&---
**********
Ingrese otra opcion
2
Mayor vida del ejercito 1 es :
Nombre: Soldado6X1 Vida: 4 Posicion: 7,i Figura: *
Nombre: Soldado8X1 Vida: 4 Posicion: 8,b Figura: *
**********
Ingrese otra opcion
3
Mayor vida del ejercito 2 es :
Nombre: Soldado2X2 Vida: 4 Posicion: 4,i Figura: &
Nombre: Soldado4X2 Vida: 4 Posicion: 3,a Figura: &
**********
```

```
Ingrese otra opcion
4
Promedio de vida del ejercito 1 es:
El promedio de vida del ejercito 1 es 2.5555555555555554
********************************
Ingrese otra opcion
5
Promedio de vida del ejercito 2 es:
El promedio de vida del ejercito 2 es 2.125
********************************
Ingrese otra opcion
6
Los datos del Ejercito 1 es :
Nombre: Soldado1X1 Vida: 2 Posicion: 3,d Figura: *
Nombre: Soldado2X1 Vida: 3 Posicion: 5,i Figura: *
Nombre: Soldado3X1 Vida: 3 Posicion: 9,e Figura: *
Nombre: Soldado4X1 Vida: 1 Posicion: 3,e Figura: *
Nombre: Soldado5X1 Vida: 3 Posicion: 8,i Figura: *
Nombre: Soldado6X1 Vida: 4 Posicion: 7,i Figura: *
Nombre: Soldado7X1 Vida: 1 Posicion: 8,a Figura: *
Nombre: Soldado8X1 Vida: 4 Posicion: 8,b Figura: *
Nombre: Soldado9X1 Vida: 2 Posicion: 5,a Figura: *
********************************
********************************
Ingrese otra opcion
7
Los datos del Ejercito 2 es :
Nombre: Soldado1X2 Vida: 1 Posicion: 9,g Figura: &
Nombre: Soldado2X2 Vida: 4 Posicion: 4,i Figura: &
Nombre: Soldado3X2 Vida: 3 Posicion: 1,c Figura: &
Nombre: Soldado4X2 Vida: 4 Posicion: 3,a Figura: &
Nombre: Soldado5X2 Vida: 1 Posicion: 3,b Figura: &
Nombre: Soldado6X2 Vida: 1 Posicion: 3,f Figura: &
Nombre: Soldado7X2 Vida: 1 Posicion: 5,c Figura: &
Nombre: Soldado8X2 Vida: 2 Posicion: 2,g Figura: &
********************************
Ingrese otra opcion
8
1er metodo de orden del Ejercito 1 :
Nombre: Soldado6X1 Vida: 4 Posicion: 7,i Figura: *
Nombre: Soldado8X1 Vida: 4 Posicion: 8,b Figura: *
Nombre: Soldado2X1 Vida: 3 Posicion: 5,i Figura: *
Nombre: Soldado3X1 Vida: 3 Posicion: 9,e Figura: *
Nombre: Soldado5X1 Vida: 3 Posicion: 8,i Figura: *
Nombre: Soldado1X1 Vida: 2 Posicion: 3,d Figura: *
Nombre: Soldado9X1 Vida: 2 Posicion: 5,a Figura: *
Nombre: Soldado4X1 Vida: 1 Posicion: 3,e Figura: *
Nombre: Soldado7X1 Vida: 1 Posicion: 8,a Figura: *
********************************
```

```
Ingrese otra opcion
9
2do metodo de orden del Ejercito 1 :
Nombre: Soldado6X1 Vida: 4 Posicion: 7,i Figura: *
Nombre: Soldado8X1 Vida: 4 Posicion: 8,b Figura: *
Nombre: Soldado5X1 Vida: 3 Posicion: 8,i Figura: *
Nombre: Soldado2X1 Vida: 3 Posicion: 5,i Figura: *
Nombre: Soldado3X1 Vida: 3 Posicion: 9,e Figura: *
Nombre: Soldado1X1 Vida: 2 Posicion: 3,d Figura: *
Nombre: Soldado9X1 Vida: 2 Posicion: 5,a Figura: *
Nombre: Soldado4X1 Vida: 1 Posicion: 3,e Figura: *
Nombre: Soldado7X1 Vida: 1 Posicion: 8,a Figura: *
**********
Ingrese otra opcion
10
1er metodo de orden del Ejercito 2 :
Nombre: Soldado2X2 Vida: 4 Posicion: 4,i Figura: &
Nombre: Soldado4X2 Vida: 4 Posicion: 3,a Figura: &
Nombre: Soldado3X2 Vida: 3 Posicion: 1,c Figura: &
Nombre: Soldado8X2 Vida: 2 Posicion: 2,g Figura: &
Nombre: Soldado1X2 Vida: 1 Posicion: 9,g Figura: &
Nombre: Soldado5X2 Vida: 1 Posicion: 3,b Figura: &
Nombre: Soldado6X2 Vida: 1 Posicion: 3,f Figura: &
Nombre: Soldado7X2 Vida: 1 Posicion: 5,c Figura: &
**********
**********
Ingrese otra opcion
11
2do metodo de orden del Ejercito 2 :
Nombre: Soldado2X2 Vida: 4 Posicion: 4,i Figura: &
Nombre: Soldado4X2 Vida: 4 Posicion: 3,a Figura: &
Nombre: Soldado3X2 Vida: 3 Posicion: 1,c Figura: &
Nombre: Soldado8X2 Vida: 2 Posicion: 2,g Figura: &
Nombre: Soldado5X2 Vida: 1 Posicion: 3,b Figura: &
Nombre: Soldado6X2 Vida: 1 Posicion: 3,f Figura: &
Nombre: Soldado7X2 Vida: 1 Posicion: 5,c Figura: &
Nombre: Soldado1X2 Vida: 1 Posicion: 9,g Figura: &
**********
Ingrese otra opcion
12
El ganador entre ambos ejercitos es :
El ejercito 1 es el ganador con un total de 23 puntos
```

```
Ingrese otra opcion
6
Los datos del Ejercito 1 es :
Nombre: Soldado1X1 Vida: 2 Posicion: 3,d Figura: *
Nombre: Soldado2X1 Vida: 3 Posicion: 5,i Figura: *
Nombre: Soldado3X1 Vida: 3 Posicion: 9,e Figura: *
Nombre: Soldado4X1 Vida: 1 Posicion: 3,e Figura: *
Nombre: Soldado5X1 Vida: 3 Posicion: 8,i Figura: *
Nombre: Soldado6X1 Vida: 4 Posicion: 7,i Figura: *
Nombre: Soldado7X1 Vida: 1 Posicion: 8,a Figura: *
Nombre: Soldado8X1 Vida: 4 Posicion: 8,b Figura: *
Nombre: Soldado9X1 Vida: 2 Posicion: 5,a Figura: *
**********
Ingrese otra opcion
7
Los datos del Ejercito 2 es :
Nombre: Soldado1X2 Vida: 1 Posicion: 9,g Figura: &
Nombre: Soldado2X2 Vida: 4 Posicion: 4,i Figura: &
Nombre: Soldado3X2 Vida: 3 Posicion: 1,c Figura: &
Nombre: Soldado4X2 Vida: 4 Posicion: 3,a Figura: &
Nombre: Soldado5X2 Vida: 1 Posicion: 3,b Figura: &
Nombre: Soldado6X2 Vida: 1 Posicion: 3,f Figura: &
Nombre: Soldado7X2 Vida: 1 Posicion: 5,c Figura: &
Nombre: Soldado8X2 Vida: 2 Posicion: 2,g Figura: &
**********
Ingrese otra opcion
13
PS F:\workspace\PracticaJava>
```