

	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 1

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	<i>Laboratorio Fundamentos de la Programación 2</i>				
TÍTULO DE LA PRÁCTICA:					
NÚMERO DE PRÁCTICA:		AÑO LECTIVO:	<i>2024</i>	NRO. SEMESTRE:	<i>2</i>
FECHA DE PRESENTACIÓN	<i>29/11/2024</i>	HORA DE PRESENTACIÓN	<i>11/59/00</i>		
INTEGRANTE (s) <i>Auccacusi Conde Brayan Carlos</i>				NOTA (0-20)	
DOCENTE(s): <i>Ing. Lino Pinto Oppe</i>					

RESULTADOS Y PRUEBAS
<p>I. EJERCICIOS RESUELTOS:</p> <p><i>El estudiante coloca la evidencia de los ejercicios propuestos realizados en la sesión de laboratorio, en el tiempo o duración indicado por el docente.</i></p> <p><i>El docente debe colocar la retroalimentación por cada ejercicio que el estudiante/grupo ha presentado</i></p> <p>CLASE PRINCIPAL</p> <pre> 1 package L8; 2 import java.util.*; 3 public class VideoJuego5 { 4 public static void main(String [] args){ 5 char decision = 'y'; 6 while(decision == 'y'){ 7 Scanner sc = new Scanner(System.in); 8 Soldado[][] campo = new Soldado[10][10]; 9 HashMap<Integer, Soldado> e1 = new HashMap<>(); 10 HashMap<Integer, Soldado> e2 = new HashMap<>(); 11 12 crearSoldados(e1, '@'); 13 crearSoldados(e2, '?'); 14 asignarSoldados(campo, e1); 15 asignarSoldados(campo, e2); 16 17 System.out.println("Leyenda nombre\n-----\n" + 18 "1erNumero = Orden de creación\n" + 19 "@ = ejército1\n? = ejército2\nÚltimo Número = Nivel de vida\n"); 20 mostrarTabla(campo); 21 </pre>

```
22     System.out.println("Mayor vida ejército 1: \n\t" + e1.get(mayorVida(e1)));
23     System.out.println("Mayor vida ejército 2: \n\t" + e2.get(mayorVida(e2)));
24
25     System.out.println("\nPromedio vida ejército 1: " + (double)sumaVida(e1)/e1.size());
26     System.out.println("Promedio vida ejército 2: " + (double)sumaVida(e2)/e2.size());
27
28     System.out.println("\nDatos en orden que fueron creados");
29     System.out.println("Ejército 1: ");
30     mostrarEnOrdenCreacion(e1);
31     System.out.println("Ejército 2: ");
32     mostrarEnOrdenCreacion(e2);
33
34     System.out.println("\nRanking de poder mayor a menor vida");
35     System.out.println("BURBUJA");
36     rankingMayorMenorBurbuja(e1);
37     mostrarSoldadosEjercito(e1, "Ejército 1: ");
38     rankingMayorMenorSeleccion(e2);
39     mostrarSoldadosEjercito(e2, "Ejército 2: ");
40
41     System.out.println("\nAleatorio");
42     aleatorio(e1);
43     aleatorio(e2);
44     mostrarSoldadosEjercito(e1, "Ejército 1: ");
45     mostrarSoldadosEjercito(e2, "Ejército 2: ");
46
47     System.out.println("\nSELECCIÓN");
48     rankingMayorMenorSeleccion(e1);
49     mostrarSoldadosEjercito(e1, "Ejército 1: ");
50     rankingMayorMenorSeleccion(e2);
51     mostrarSoldadosEjercito(e2, "Ejército 2: ");
52
53     System.out.println();
54     determinarGanador(e1, e2);
55
56     System.out.print("¿Desea iniciar un nuevo juego? (y/n): ");
57     decision = sc.nextLine().charAt(0);
58 }
59
60 }
61
62 public static void crearSoldados(HashMap<Integer, Soldado> ejercito, char a){
63     Random random = new Random();
64     /*Asignamos soldados al HashMap
65     * La cantidad de soldados es determinada aleatoriamente
66     * Damos nombre a cada soldado creado, considerando el ejército al que pertenece (@ o ?)
67     */
68     for(int i = 0; i < random.nextInt(10) + 1; i++){
69         Soldado aux = new Soldado();
70         aux.setVida(random.nextInt(5) + 1);
71         aux.setNombre(i + "." + a + aux.getVida());
72         ejercito.put(i, aux);
73     }
74 }
```

```
75
76 public static void mostrarEnOrdenCreacion(HashMap<Integer, Soldado> ejercito){
77     for(int i = 0; i < ejercito.size(); i++){
78         for(int j = 0; j < ejercito.size(); j++){
79             //Comparamos el primer carácter de cada soldado, este nos dice su orden de creación.
80             if(ejercito.get(j).getNombre().charAt(0) == i + '0'){
81                 System.out.println(ejercito.get(j));
82             }
83         }
84     }
85 }
86
87 /*mostrarSoldadosEjercito muestra los valores del HashMap sin importar si se hicieron
88 cambios en los "valores" del HashMap */
89 public static void mostrarSoldadosEjercito(HashMap<Integer, Soldado> ejercito, String ejercitoNombre){
90     System.out.println(ejercitoNombre);
91     for(int key : ejercito.keySet()){
92         System.out.println(ejercito.get(key));
93     }
94 }
95
96 public static void asignarSoldados(Soldado[][] campo, HashMap<Integer, Soldado> ejercito){
97     Random rand = new Random();
98     int fila, columna;
99     for (int i = 0; i < ejercito.size(); i++){
100         boolean aux = true;
101         while(aux){
102             fila = rand.nextInt(10);
103             columna = rand.nextInt(10);
104             /*Si campo está vacío le asignamos un elemento del HashMap (correspondiente a un ejército determinado) */
105             if(campo[fila][columna] == null){
106                 campo[fila][columna] = ejercito.get(i);
107                 ejercito.get(i).setFila(fila);
108                 ejercito.get(i).setColumna(columna);
109                 aux = false;
110             }
111         }
112     }
113 }
114
115 public static void mostrarTabla(Soldado[][] campo){
116     System.out.print(" ");
117     for(char i = 'A'; i < 'K'; i++){
118         System.out.print(" " + i + " ");
119     }
120     System.out.println();
121     for(int i = 0; i < campo.length; i++){
122         if(i == 9)
123             System.out.print(i + 1);
124         else
125             System.out.print((i + 1) + " ");
126
127         for(int j = 0; j < campo[i].length; j++){
128             if (campo[i][j] != null)
129                 System.out.print("|" + campo[i][j].getNombre());
130             else
131                 System.out.print("|   ");
132         }
133         System.out.println(" |");
134         System.out.println(" -----");
135     }
136 }
```

```
137
138 public static int mayorVida(HashMap<Integer, Soldado> ejercito){
139     int indexMayor = 0;
140     for(int i = 0; i < ejercito.size(); i++){
141         if(ejercito.get(i).getVida() > ejercito.get(indexMayor).getVida()){
142             indexMayor = i;
143         }
144     }
145     return indexMayor;
146 }
147
148 public static void aleatorio(HashMap<Integer, Soldado> ejercito){
149     /*Generamos 2 números aleatorios que actuarán como índices, luego intercambiamos los
150     * "valores" de cada entrada del HashMap con estos índices.
151     */
152     Random rand = new Random();
153     Soldado aux;
154     int r1, r2;
155     for(int i = 0; i < ejercito.size(); i++){
156         r1 = rand.nextInt(ejercito.size());
157         r2 = rand.nextInt(ejercito.size());
158         aux = ejercito.get(r1);
159         ejercito.put(r1, ejercito.get(r2));
160         ejercito.put(r2, aux);
161     }
162 }
163
164 public static void rankingMayorMenorBurbuja(HashMap<Integer, Soldado> ejercito) {
165     for (int i = 0; i < ejercito.size(); i++) {
166         /* Cogemos el primer elemento y lo comparamos con los demás, para las siguientes pasadas
167         ya no consideramos el primero puesto que ya es el mayor */
168         for (int j = i + 1; j < ejercito.size(); j++) {
169             if (ejercito.get(j).getVida() > ejercito.get(i).getVida()) {
170                 // Intercambiar posiciones para burbuja
171                 Soldado temp = ejercito.get(i);
172                 ejercito.put(i, ejercito.get(j));
173                 ejercito.put(j, temp);
174             }
175         }
176     }
177 }
178
179 public static void rankingMayorMenorSeleccion(HashMap<Integer, Soldado> ejercito){
180     for (int i = 0; i < ejercito.size() - 1; i++) {
181         int indexMayor = i;
182         for (int j = i + 1; j < ejercito.size(); j++) {
183             if (ejercito.get(j).getVida() > ejercito.get(indexMayor).getVida()) {
184                 indexMayor = j;
185             }
186         }
187         // Intercambiar los soldados
188         Soldado temp = ejercito.get(i);
189         ejercito.put(i, ejercito.get(indexMayor));
190         ejercito.put(indexMayor, temp);
191     }
192 }
```

```
193
194 public static int sumaVida(HashMap<Integer, Soldado> ejercito){
195     int sumaVida = 0;
196     for(int i = 0; i < ejercito.size(); i++){
197         sumaVida += ejercito.get(i).getVida();
198     }
199     return sumaVida;
200 }
201
202 public static void determinarGanador(HashMap<Integer, Soldado> ejercito1, HashMap<Integer, Soldado> ejercito2){
203     /*El ganador se determina de acuerdo a la suma total de las vidas de los soldados de cada ejército */
204     int vidaTotalEjer1, vidaTotalEjer2;
205     vidaTotalEjer1 = sumaVida(ejercito1);
206     vidaTotalEjer2 = sumaVida(ejercito2);
207
208     if (vidaTotalEjer1 == vidaTotalEjer2) {
209         System.out.println("EMPATE\nEjército1 = " + vidaTotalEjer1 +
210             " ---- Ejército2 = " + vidaTotalEjer2);
211     }
212     else if (vidaTotalEjer1 > vidaTotalEjer2) {
213         System.out.println("GANA EJÉRCITO 1\nEjército1 = " + vidaTotalEjer1 +
214             " > Ejército2 = " + vidaTotalEjer2);
215     }
216     else{
217         System.out.println("GANA EJÉRCITO 2\nEjército1 = " + vidaTotalEjer1 +
218             " < Ejército2 = " + vidaTotalEjer2);
219     }
220 }
221 }
222
```

CLASE SOLDADO

```
1 package _8;  
2  
3 public class Soldado {  
4     private String nombre;  
5     private int vida;  
6     private int fila;  
7     private int columna;  
8  
9     public void setNombre( String n){  
10         nombre = n;  
11     }  
12     public void setFila(int f){  
13         fila = f;  
14     }  
15     public void setColumna(int c){  
16         columna = c;  
17     }  
18     public void setVida(int v){  
19         vida = v;  
20     }  
21  
22     public String getNombre(){  
23         return nombre;  
24     }  
25     public int getFila(){  
26         return fila;  
27     }  
28     public int getColumna(){  
29         return columna;  
30     }  
31     public int getVida(){  
32         return vida;  
33     }  
34  
35     public String toString(){  
36         return "Nombre: " + getNombre() + " Vida: " + getVida() +  
37             " Fila " + (getFila() + 1) + " Columna: " + getColumna();  
38     }  
39 }
```

II. PRUEBAS

```
nts\BRAYAN\FP2-LABORATORIOS\bin\ 'L8.VideoJuego5'
Leyenda nombre
-----
1erNumero = Orden de creación
@ = ejército1
? = ejército2
Último Número = Nivel de vida

  A   B   C   D   E   F   G   H   I   J
1 |   |   |   |   |   |   |   |   |
  -----
2 |   |   |   |   |   | 3.@1 |   |   |
  -----
3 |   |   |   |   |   |   |   |   |   |
  -----
4 |   |   |   |   |   |   |   |   |   |
  -----
5 |   |   |   |   |   |   |   |   |   |
  -----
6 |   |   |   |   |   |   |   |   |   |
  -----
7 |   |   |   |   |   |   |   |   |   |
  -----
8 | 2.?5|1.?4|   | 0.@1|5.?1|   |   |   |   |
  -----
9 |   |   |   |   |   |   | 4.?1|0.?2|   | 1.@3 |
  -----
10|   |   |   |   |   |   |   | 3.?3|2.@4|   |
  -----

-----
Mayor vida ejército 1:
      Nombre: 2.@4   Vida: 4   Fila 10   Columna: 8
Mayor vida ejército 2:
      Nombre: 2.?5   Vida: 5   Fila 8     Columna: 0

Promedio vida ejército 1: 2.25
Promedio vida ejército 2: 2.6666666666666665
```


Datos en orden que fueron creados

Ejército 1:

Nombre: 0.@1	Vida: 1	Fila 8	Columna: 3
Nombre: 1.@3	Vida: 3	Fila 9	Columna: 9
Nombre: 2.@4	Vida: 4	Fila 10	Columna: 8
Nombre: 3.@1	Vida: 1	Fila 2	Columna: 6

Ejército 2:

Nombre: 0.?2	Vida: 2	Fila 9	Columna: 7
Nombre: 1.?4	Vida: 4	Fila 8	Columna: 1
Nombre: 2.?5	Vida: 5	Fila 8	Columna: 0
Nombre: 3.?3	Vida: 3	Fila 10	Columna: 7
Nombre: 4.?1	Vida: 1	Fila 9	Columna: 6
Nombre: 5.?1	Vida: 1	Fila 8	Columna: 4

Ranking de poder mayor a menor vida

BURBUJA

Ejército 1:

Nombre: 2.@4	Vida: 4	Fila 10	Columna: 8
Nombre: 1.@3	Vida: 3	Fila 9	Columna: 9
Nombre: 0.@1	Vida: 1	Fila 8	Columna: 3
Nombre: 3.@1	Vida: 1	Fila 2	Columna: 6

Ejército 2:

Nombre: 2.?5	Vida: 5	Fila 8	Columna: 0
Nombre: 1.?4	Vida: 4	Fila 8	Columna: 1
Nombre: 3.?3	Vida: 3	Fila 10	Columna: 7
Nombre: 0.?2	Vida: 2	Fila 9	Columna: 7
Nombre: 4.?1	Vida: 1	Fila 9	Columna: 6
Nombre: 5.?1	Vida: 1	Fila 8	Columna: 4

Aleatorio

Ejército 1:

Nombre: 2.@4	Vida: 4	Fila 10	Columna: 8
Nombre: 3.@1	Vida: 1	Fila 2	Columna: 6
Nombre: 1.@3	Vida: 3	Fila 9	Columna: 9
Nombre: 0.@1	Vida: 1	Fila 8	Columna: 3

Ejército 2:

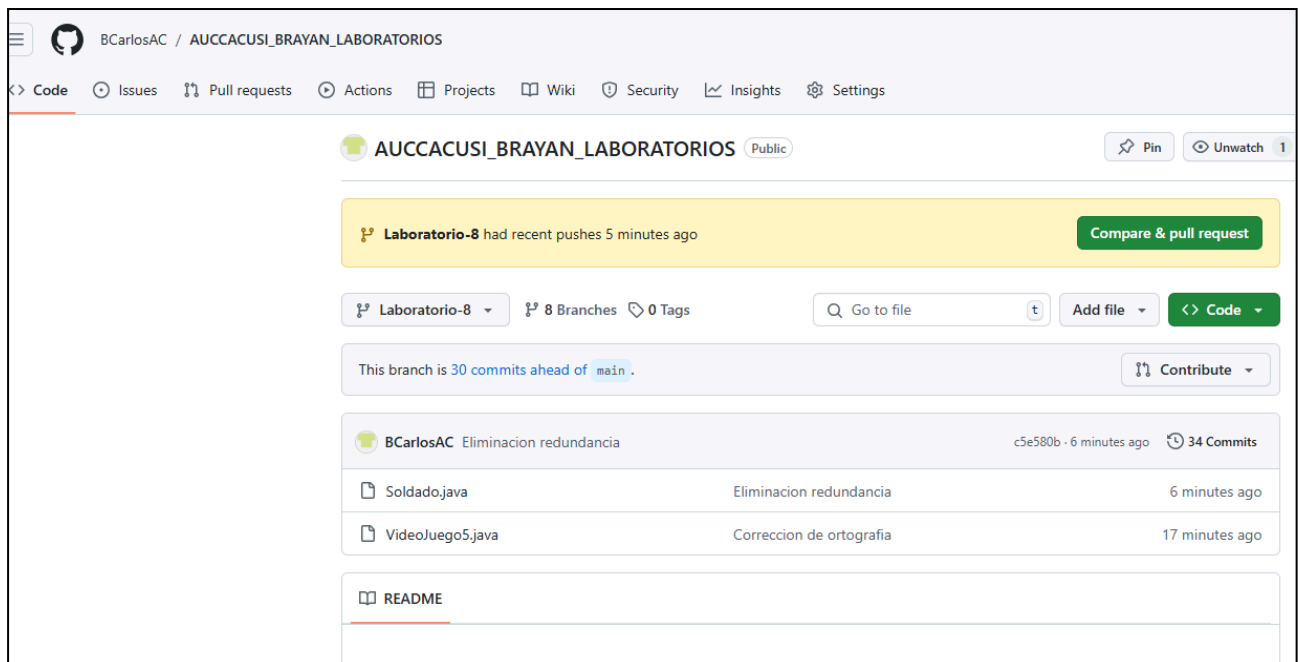
Nombre: 5.?1	Vida: 1	Fila 8	Columna: 4
Nombre: 1.?4	Vida: 4	Fila 8	Columna: 1
Nombre: 3.?3	Vida: 3	Fila 10	Columna: 7
Nombre: 0.?2	Vida: 2	Fila 9	Columna: 7
Nombre: 2.?5	Vida: 5	Fila 8	Columna: 0
Nombre: 4.?1	Vida: 1	Fila 9	Columna: 6


```

SELECCIÓN
Ejército 1:
Nombre: 2.@4      Vida: 4      Fila 10      Columna: 8
Nombre: 1.@3      Vida: 3      Fila 9        Columna: 9
Nombre: 3.@1      Vida: 1      Fila 2        Columna: 6
Nombre: 0.@1      Vida: 1      Fila 8        Columna: 3
Ejército 2:
Nombre: 2.?5      Vida: 5      Fila 8        Columna: 0
Nombre: 1.?4      Vida: 4      Fila 8        Columna: 1
Nombre: 3.?3      Vida: 3      Fila 10       Columna: 7
Nombre: 0.?2      Vida: 2      Fila 9        Columna: 7
Nombre: 5.?1      Vida: 1      Fila 8        Columna: 4
Nombre: 4.?1      Vida: 1      Fila 9        Columna: 6

GANA EJÉRCITO 2
Ejército1 = 9 < Ejército2 = 16
¿Desea iniciar un nuevo juego? (y/n): n
PS C:\Users\Hogar\Documents\BRAYAN\FP2-LABORATORIOS>
  
```


COMMITTS



The screenshot shows the GitHub interface for the repository **AUCCACUSI_BRAYAN_LABORATORIOS**. The repository is public and has 8 branches and 0 tags. The current branch is **Laboratorio-8**, which is 30 commits ahead of the **main** branch. The repository was last pushed 5 minutes ago by **Laboratorio-8**. The commit history shows two recent commits by **BCarlosAC**: "Eliminacion redundancia" (6 minutes ago) and "Correccion de ortografia" (17 minutes ago). The repository also has a README file.

Commits on Nov 26, 2024		
Delete Soldado.java	Verified	3db30ba <>
BCarlosAC authored 3 days ago		
Delete VideoJuego4.java	Verified	bb15287 <>
BCarlosAC authored 3 days ago		
Pasamos codigos del lab7 a lab8 para trabajar		c6608f1 <>
BCarlosAC committed 3 days ago		
Commits on Nov 29, 2024		
Creacion de HashMap y modificacion del metodo crearSoldados		4ff76f8 <>
BCarlosAC committed 12 hours ago		
Modificacion del metodo asignarSoldados, correccion en crear soldados		462f170 <>
BCarlosAC committed 12 hours ago		
Metodo mayorVida, modificacion de la clave del HashMap		8acfb42 <>
BCarlosAC committed 12 hours ago		
Modificacion metodo rankingMayorMenorBurbuja		d0e7da2 <>
BCarlosAC committed 1 hour ago		
Modificacion metodos ordenar de mayor a menor por Burbuja y Seleccion		4654d25 <>
BCarlosAC committed in 1 hour		
Modificacion metodos de ordenamiento para que modifiquen el Hashmap s...		77636c6 <>
BCarlosAC committed in 2 hours		
Metodos mostrarEnOrdenCreacion y mostrarSoldadosEjercito		8cb5f78 <>
BCarlosAC committed in 2 hours		

BCarlosAC committed in 2 hours		
Metodos mostrarEnOrdenCreacion y mostrarSoldadosEjercito		8cb5f78 <>
BCarlosAC committed in 2 hours		
Metodo aleatorio y adaptacion de los metodos en el main		cfc53e1 <>
BCarlosAC committed 2 hours ago		
Metodo determinarGanador		93943b5 <>
BCarlosAC committed 2 hours ago		
Correccion de errores		53d7ca2 <>
BCarlosAC committed 1 hour ago		
Optimizacion del codigo		d0af64b <>
BCarlosAC committed 27 minutes ago		
Correccion de ortografia		f5ecb73 <>
BCarlosAC committed 20 minutes ago		
Eliminacion redundancia		c5e580b <>
BCarlosAC committed 9 minutes ago		

 © 2024 GitHub, Inc.
 Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

LINK REPOSITORIO:

https://github.com/BCarlosAC/AUCCACUSI_BRAYAN_LABORATORIOS.git

¿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?

- *El código no necesitaba entrada a excepción de decidir iniciar un nuevo juego por lo que solo probe con eso.*
- *Cuando ejecutaba el código esperaba que saliera mi campo de batalla junto con los soldados y la ejecución de los métodos que puse en el main.*
- *Obtuve los métodos así como los print() de mi main.*

III. CUESTIONARIO:

Colocar la evidencia de las respuestas realizadas al cuestionario enunciado en la guía práctica de laboratorio.

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	1	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos 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		18	

Tabla 2: Rúbrica para contenido del Informe y demostración

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 12</p>

CONCLUSIONES

Los HashMap son muy útiles pues permiten tener como llaves no solo a índices sino a cualquier otro tipo de dato siempre que se a una clase (como String), en este ejercicio los usamos para almacenar a nuestro soldados por lo que son útiles para almacenar eso tambien, los HashMap son muy similares a los ArrayList.

METODOLOGÍA DE TRABAJO

- *Copie mi código anterior para trabajar teniendo una base.*
- *Inicie mi git y lo conecte con mi repositorio remoto.*
- *Comencé con la creación de mis variables de referencia a mis HashMap, modifique los diferentes métodos adaptandolos a los HashMap.*
- *Ejecute mi código y vi que errores había para luego corregirlos o tomar otra lógica.*
- *Verifique si habia codigo innecesario o redundante (optimización).*
- *Puse comentarios en cada parte necesaria de mi codigo y finalmente corregí las faltas ortograficas.*

REFERENCIAS Y BIBLIOGRAFÍA

Aedo López, M. (2024). ArrayList y HashMap. EPIS UNSA.