

# Informe de Laboratorio 14

## Tema: Reinos y ejercitos

Nota

Estudiante	Escuela	Asignatura
Sarayasi Huanaco,Jeferson Jesus	Escuela Profesional de Ingeniería de Sistemas	Fundamentos de la Programación 2

Laboratorio	Tema	Duración
14	Reinos y ejercitos	48 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - B		

Docente
Aedo Lopez, Marco Wilfredo

### 1. Tarea

### 2. Equipos, Materiales y temas utilizados

- Sistema Operativo Windows 11.
- VIM 9.0.
- OpenJHK 64-Bits 17.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional
- Arreglos Estandar

### 3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar
- <https://github.com/JefersonSH/FP2-2024B.git>
- URL para el laboratorio 07 en el Repositorio GitHub.

## 4. Ejercicios Resuletos

### Ejercicio 01

- **Commit 1:** VideoJuego numero 11 con las modificaciones pedidas por el docente implementando los reinos y los ejércitos

Listing 1: Clase VideoJuego

```
1 import java.util.*;
2
3 public class VideoJuego11 {
4     static Scanner sc = new Scanner(System.in);
5     private static int maxSoldados = 10;
6     private static int maxEjercitos = 10;
7     private static int maxReinos = 2;
8     private static int ladoTablero = 10;
9     private static ArrayList<Integer> filas = new ArrayList<>();
10    private static ArrayList<Integer> columnas = new ArrayList<>();
11
12    public static void main(String[] args){
13        Menu();
14    }
15
16    public static void Menu(){
17        ArrayList<Reino> reinos = new ArrayList<>();
18        CrearEjercitos(reinos);
19        Jugar(reinos);
20    }
21
22    public static void CrearEjercitos(ArrayList<Reino> reinos){
23        String[] nombreReinos = {"Inglaterra", "Francia", "Sacro Imperio", "Castilla -
24                                Aragon", "Moros"};
25        int random1, random2;
26        // Seleccionando el nombre de 2 Reinos diferentes aleatorios
27        while(true){
28            random1 = (int)(Math.random() * nombreReinos.length);
29            random2 = (int)(Math.random() * nombreReinos.length);
30            if(random1 != random2) break;
31        }
32        // Creando los 2 reinos
33        reinos.add(new Reino(nombreReinos[random1]));
34        reinos.add(new Reino(nombreReinos[random2]));
35        // Creando de 1 a 10 ejrcitos para cada reino
36        for(Reino reino: reinos){
37            for(int i = 0; i < (int)(Math.random() * maxEjercitos + 1); i++){
38                int fila = 0, columna = 0, contador = 0;
39                Ejercito ejercito = new Ejercito(i + 1, reino.getLetra() + (i + 1),
40                    reino.getNombre());
41                while(true){
42                    contador = 0;
43                    ejercito.setFila((int)(Math.random() * ladoTablero + 1));
44                    ejercito.setColumna((int)(Math.random() * ladoTablero + 1));
45                    if(Verificar(ejercito.getFila(), ejercito.getColumna())) contador++;
46                    // Si el contador es ms de 0, significa que hay al menos 1 coincidencia de
47                    // ejrcitos con la misma fila y columna, por ende, se generan otros
48                    // valores para fila y columna
```

```
45         if(contador == 0) break;
46     }
47     reino.getEjercitos().add(ejercito);    // Agregando los datos del ejrcito al
48         reino
49     Actualizar(reinos);
50 }
51 }
52
53 public static void Actualizar(ArrayList<Reino> reinos){
54     filas.clear();
55     columnas.clear();
56     for(Reino reino : reinos)
57         for(Ejercito ejercito : reino.getEjercitos()){
58             filas.add(ejercito.getFila());
59             columnas.add(ejercito.getColumna());
60         }
61 }
62
63 public static boolean Verificar(int fila, int columna){
64     for(int i = 0; i < filas.size(); i++)
65         if(filas.get(i) == fila && columnas.get(i) == columna) return true;
66     return false;
67 }
68
69 public static void ImprimirReinos(ArrayList<Reino> reinos){
70     System.out.println("----- SOLDADOS EN TODOS LOS REINOS -----");
71     // Recorriendo todos los reinos, ejrcitos y soldados
72     for (Reino reino : reinos){
73         System.out.println("Reino: " + reino.getNombre() + "(" + reino.getLetra() + ")");
74         for (Ejercito ejercito : reino.getEjercitos()){
75             System.out.println("- Ejrcito: " + ejercito.getNombre() + "\tPosicion: " +
76                 ejercito.getFila() + "x" + ejercito.getColumna());
77         }
78     }
79
80 public static void ImprimirTableroEjercitos(ArrayList<Reino> reinos) {
81     for (int i = 0; i < ladoTablero; i++) {
82         System.out.println("\n+-----+-----+-----+-----+");
83         System.out.print("|");
84         for (int j = 0; j < ladoTablero; j++) {
85             boolean encontrado = false;
86             for (Reino reino : reinos) {
87                 for (Ejercito ejercito : reino.getEjercitos()) {
88                     if (ejercito.getFila() == (i + 1) && ejercito.getColumna() == (j + 1)) {
89                         System.out.print(" " + reino.getLetra() + " ");
90                         encontrado = true;
91                         break; // Salir del bucle de ejrcitos si se ha encontrado y
92                             mostrado el ejrcito
93                     }
94                 }
95             }
96             if (encontrado) {
97                 break; // Salir del bucle de reinos si se encontr el ejrcito
98             }
99         }
100     }
101 }
```

```

98         if (!encontrado) {
99             System.out.print(" "); // Espacios si no se encontr ningn ejrcito en esa
                posicin
100         }
101         System.out.print("|");
102     }
103 }
104 System.out.println("\n+---+---+---+---+---+---+---+---+---+---\n");
105 }
106
107 public static void JugarAutomaticamente(ArrayList<Ejercito> ejercitos){
108     int[] vidaTotal = new int[2];
109     int contador = 0;
110     CrearSoldados(ejercitos);
111     for(Ejercito ejercito: ejercitos){
112         for(Soldado soldado: ejercito.getSoldados()){
113             vidaTotal[contador] += soldado.getVidaActual();
114         }
115         contador++;
116     }
117     System.out.println("Vida Total del Ejercito " + ejercitos.get(0).getNombre() + ": " +
        vidaTotal[0] + "\nVida Total del Ejercito " + ejercitos.get(1).getNombre() + ": " +
        vidaTotal[1]);
118     if(vidaTotal[0] > vidaTotal[1]){
119         ejercitos.get(1).getSoldados().clear();
120         System.out.println("Se termino el enfrentamiento. Ganador: " +
            ejercitos.get(0).getNombre());
121     } else {
122         ejercitos.get(0).getSoldados().clear();
123         System.out.println("Se termino el enfrentamiento. Ganador: " +
            ejercitos.get(1).getNombre());
124     }
125 }
126
127 public static void Combate(int fila, int columna, Ejercito ejercitoActual, String
    direccion, ArrayList<Reino> reinos) {
128     boolean ejercitoEncontrado = false; // Bandera para verificar si se encontr un ejrcito
        en la posicin deseada
129     ArrayList<Ejercito> ejercitos = new ArrayList<>();
130     int contador = reinos.size() - 1;
131     for (Reino reino : reinos) {
132         for (Ejercito ejercito : reino.getEjercitos()) {
133             if (ejercito.getFila() == fila && ejercito.getColumna() == columna &&
                !ejercito.getNombre().equals(ejercitoActual.getNombre()) &&
                !ejercito.getReino().equals(ejercitoActual.getReino())) {
134                 //Se cruzan 2 ejércitos de diferentes reinos
135                 ejercitos.add(ejercito);
136                 ejercitos.add(ejercitoActual);
137                 System.out.println("\n***** QUE DESEA REALIZAR *****\n1. Jugar
                    Manualmente\t2. Jugar Automaticamente");
138                 String opcion = sc.nextLine();
139                 if(opcion.equals("1")) Jugar2(ejercitos, reino);
140                 else JugarAutomaticamente(ejercitos);
141                 ejercitoEncontrado = true;
142                 if(ejercitos.get(0).getSoldados().size() == 0){
143                     System.out.println("El ejercito rival esta vacio");

```

```
144         ejercitos.get(1).setFila(fila);
145         ejercitos.get(1).setColumna(columna);
146         reino.getEjercitos().remove(ejercitos.get(0));
147         System.out.println("El ejercito rival esta vacio2");
148     }
149     if(ejercitos.get(1).getSoldados().size() == 0){
150         System.out.println("El ejercito actual esta vacio");
151         reinos.get(contador).getEjercitos().remove(ejercitos.get(1));
152         System.out.println("El ejercito actual esta vacio2");
153     }
154     break;
155 } else if (ejercito.getFila() == fila && ejercito.getColumna() == columna &&
        !ejercito.getNombre().equals(ejercitoActual.getNombre()) &&
        ejercito.getReino().equals(ejercitoActual.getReino())) {
156     System.out.println("#####\nAccion no valida, no se permiten enfrentamientos
        locales\n#####");
157     ejercitoEncontrado = true;
158     break;
159 }
160 }
161 if (ejercitoEncontrado) {
162     break;
163 }
164 contador--;
165 }
166 if (!ejercitoEncontrado) {
167     ejercitoActual.setFila(fila);
168     ejercitoActual.setColumna(columna);
169 }
170 }
171
172 public static boolean CumplesCondicion(String t, String direccion, ArrayList<Reino> reinos,
        String reinoActual){
173     int fila = Integer.parseInt(t.substring(0,t.indexOf("x")));
174     int columna = Integer.parseInt(t.substring(t.indexOf("x") + 1, t.length()));
175     for(Reino reino : reinos)
176         for(Ejercito ejercito : reino.getEjercitos())
177             if(ejercito.getFila() == fila && ejercito.getColumna() == columna &&
                reino.getNombre().equals(reinoActual)){
178                 if(direccion.equals("w") && ejercito.getFila() - 2 >= 0){
179                     Combate(fila - 1, columna, ejercito, direccion, reinos);
180                     return true;
181                 } else if(direccion.equals("s") && ejercito.getFila() < ladoTablero){
182                     Combate(fila + 1, columna, ejercito, direccion, reinos);
183                     return true;
184                 } else if(direccion.equals("a") && ejercito.getColumna() - 2 >= 0){
185                     Combate(fila, columna - 1, ejercito, direccion, reinos);
186                     return true;
187                 } else if(direccion.equals("d") && ejercito.getColumna() < ladoTablero){
188                     Combate(fila, columna + 1, ejercito, direccion, reinos);
189                     return true;
190                 } else{
191                     System.out.println("Direccion invalida, por favor ingrese los valores de
                        nuevo");
192                     return false;
193                 }
194             }
```

```
194     }
195     System.out.println("Ejercito no encontrado, por favor ingrese los datos de nuevo");
196     return false;
197 }
198
199 public static void Movimiento(ArrayList<Reino> reinos, int turno){
200     String jugador = "", posicion = "", direccion = "";
201     System.out.println("Jugador 1: " + reinos.get(0).getNombre() + "\tJugador2: " +
202         reinos.get(1).getNombre());
203     System.out.println("Ingrese la posicion del ejercito a mover. Ejemplo (1x2)");
204
205     turno %= 2;
206     if(turno == 0) jugador = reinos.get(0).getNombre();
207     if(turno == 1) jugador = reinos.get(1).getNombre();
208     System.out.println("Reino Actual: " + jugador);
209
210     while(true){ //Verifica que los datos ingresados sean validos, tanto el ejercito como
211         la direccion elegida
212         posicion = sc.nextLine();
213         System.out.println("Ingrese la direccion de movimiento ( w: arriba; s: abajo; a:
214             izquierda; d: derecha)");
215         direccion = sc.nextLine();
216         if(CumpleCondicion(posicion, direccion, reinos, jugador)) break;
217     }
218 }
219
220 public static void Jugar(ArrayList<Reino> reinos){
221     ImprimirReinos(reinos);
222     ImprimirTableroEjercitos(reinos);
223     int turno = 0;
224     while(true){
225         Movimiento(reinos, turno);
226         turno++;
227         ImprimirTableroEjercitos(reinos);
228         if(reinos.get(0).getEjercitos().size() == 0){
229             System.out.println("Se termino el juego. Ganador: " + reinos.get(1).getNombre());
230             break;
231         }
232         if(reinos.get(1).getEjercitos().size() == 0){
233             System.out.println("Se termino el juego. Ganador: " + reinos.get(0).getNombre());
234             break;
235         }
236     }
237 }
238
239 public static void Actualizar2(ArrayList<Ejercito> ejercitos){
240     filas.clear();
241     columnas.clear();
242     for(Ejercito ejercito : ejercitos)
243         for(Soldado soldado : ejercito.getSoldados()){
244             filas.add(soldado.getFila());
245             columnas.add(soldado.getColumna());
246         }
247     }
248 }
249
250 public static void ImprimirTableroSoldados(ArrayList<Ejercito> ejercitos) {
```

```
247     for (int i = 0; i < ladoTablero; i++) {
248         int cont = 1;
249         System.out.println("\n+---+---+---+---+---+---+---+---+---+");
250         System.out.print("|");
251         for (int j = 0; j < ladoTablero; j++) {
252             cont = 1;
253             boolean encontrado = false;
254             for (Ejercito ejercito: ejercitos) {
255                 for (Soldado soldado : ejercito.getSoldados()) {
256                     if (soldado.getFila() == (i + 1) && soldado.getColumna() == (j + 1)) {
257                         System.out.print(" " + cont + " ");
258                         encontrado = true;
259                         break; // Salir del bucle de ejrcitos si se ha encontrado y
                             mostrado el ejrcito
260                     }
261                 }
262                 if (encontrado) {
263                     break; // Salir del bucle de reinos si se encontr el ejrcito
264                 }
265                 cont = 2;
266             }
267             if (!encontrado) {
268                 System.out.print(" "); // Espacios si no se encontr ningn ejrcito en esa
                             posicin
269             }
270             System.out.print("|");
271         }
272     }
273     System.out.println("\n+---+---+---+---+---+---+---+---+---+");
274 }
275
276 public static void CrearSoldados(ArrayList<Ejercito> ejercitos){
277     for(Ejercito ejercito: ejercitos){
278         for(int i = 0; i < (int)(Math.random() * maxSoldados + 1); i++){
279             int fila = 0, columna = 0, contador = 0;
280             while(true){
281                 contador = 0;
282                 fila = (int)(Math.random() * ladoTablero + 1);
283                 columna = (int)(Math.random() * ladoTablero + 1);
284                 if(Verificar(fila, columna)) contador++;
285                 // Si el contador es ms de 0, significa que hay al menos 1 coincidencia de
                             ejrcitos con la misma fila y columna, por ende, se generan otros valores
                             para fila y columna
286                 if(contador == 0) break;
287             }
288             int vida = (int)(Math.random() * 5 + 1), ataque = (int)(Math.random() * 5 + 1),
                defensa = (int)(Math.random() * 5 + 1), velocidad = (int)(Math.random() * 5 +
                1);
289             Soldado soldado = new Soldado("Soldado " + ejercito.getEjercito() + "x" + (i + 1),
                vida, ejercito.getEjercito(), fila, columna, ataque, defensa, velocidad);
290             ejercito.getSoldados().add(soldado); // Agregando los datos del ejrcito al reino
291             Actualizar2(ejercitos);
292         }
293     }
294 }
295
```

```
296 public static void Combate2(int fila, int columna, Soldado soldadoActual, int index,
297     String direccion, String ejercitoActual, ArrayList<Ejercito> ejercitos){
298     int contador = 2;
299     boolean soldadoEncontrado = false; // Bandera para verificar si se encontr un ejrcito
300     // en la posicin deseada
301     for(Ejercito ejercito : ejercitos){
302         contador--;
303         for(Soldado soldado : ejercito.getSoldados()){
304             if(soldado.getFila() == fila && soldado.getColumna() == columna &&
305                 !soldado.getNombre().equals(soldadoActual.getNombre()) &&
306                 !ejercito.getNombre().equals(ejercitoActual)){
307                 //Empieza el combate entre soldados
308                 int num = (int)(Math.random() * 100 + 1);
309                 double total = soldadoActual.getVidaActual() + soldado.getVidaActual();
310                 double probabilidadA = 100 * (soldadoActual.getVidaActual() / total);
311                 System.out.println("Probabilidades de ganar:\nSoldado actual: " + probabilidadA
312                     + "% --- Soldado enemigo: " + (100 - probabilidadA) + "%");
313                 if(num < probabilidadA){ // El ganador seria el soldado actual
314                     ejercitos.get(contador).getSoldados().get(index).cambiarPosicion(soldadoActual.getVidaActual()
315                         + 1, soldado.getFila(), soldado.getColumna());
316                     ejercito.getSoldados().remove(soldado);
317                     System.out.println("El Soldado actual gana");
318                 } else { // Si no el ganador seria el soldado rival
319                     soldado.setVidaActual(soldado.getVidaActual() + 1);
320                     ejercitos.get(contador).getSoldados().remove(index);
321                     System.out.println("El Soldado enemigo gana");
322                 }
323                 soldadoEncontrado = true;
324                 break;
325             } else if(soldado.getFila() == fila && soldado.getColumna() == columna &&
326                 !soldado.getNombre().equals(soldadoActual.getNombre()) &&
327                 !ejercito.getNombre().equals(ejercitoActual)){
328                 System.out.println("Accion no valida, no se permiten enfrentamientos locales");
329                 soldadoEncontrado = true;
330                 break;
331             }
332         }
333     }
334     if (soldadoEncontrado) {
335         break;
336     }
337     if (!soldadoEncontrado) {
338         soldadoActual.setFila(fila);
339         soldadoActual.setColumna(columna);
340     }
341 }
```

```
342 public static boolean CumpleCondicion2(String t, String direccion, ArrayList<Ejercito>
343     ejercitos, String ejercitoActual){
344     int fila = Integer.parseInt(t.substring(0,t.indexOf("x")));
345     int columna = Integer.parseInt(t.substring(t.indexOf("x") + 1, t.length()));
346     for(Ejercito ejercito : ejercitos)
347         for(Soldado soldado : ejercito.getSoldados())
348             if(soldado.getFila() == fila && soldado.getColumna() == columna &&
349                 ejercito.getNombre().equals(ejercitoActual)){
350                 if(direccion.equals("w") && soldado.getFila() - 2 >= 0){
```



```
342         Combate2(fila - 1, columna, soldado,
343                 ejercito.getSoldados().indexOf(soldado), direccion,
344                 ejercito.getNombre(), ejercitos);
345         return true;
346     } else if(direccion.equals("s") && soldado.getFila() < ladoTablero){
347         Combate2(fila + 1, columna, soldado,
348                 ejercito.getSoldados().indexOf(soldado), direccion,
349                 ejercito.getNombre(), ejercitos);
350         return true;
351     } else if(direccion.equals("a") && soldado.getColumna() - 2 >= 0){
352         Combate2(fila, columna - 1, soldado,
353                 ejercito.getSoldados().indexOf(soldado), direccion,
354                 ejercito.getNombre(), ejercitos);
355         return true;
356     } else if(direccion.equals("d") && ejercito.getColumna() < ladoTablero){
357         Combate2(fila, columna + 1, soldado,
358                 ejercito.getSoldados().indexOf(soldado), direccion,
359                 ejercito.getNombre(), ejercitos);
360         return true;
361     } else{
362         System.out.println("Direccion invalida, por favor ingrese los valores de
363                             nuevo");
364         return false;
365     }
366 }
367 }
368 System.out.println("Soldado no encontrado, por favor ingrese los datos de nuevo");
369 return false;
370 }
371
372 public static void Movimiento2(ArrayList<Ejercito> ejercitos, int turno){
373     String jugador = "", posicion = "", direccion = "";
374     System.out.println("Jugador 1: Ejercito " + ejercitos.get(0).getNombre() + "\tJugador2:
375                         Ejercito " + ejercitos.get(1).getNombre());
376     System.out.println("Ingrese la posicion del soldado a mover. Ejemplo (1x2)");
377
378     turno %= 2;
379     if(turno == 0) jugador = ejercitos.get(0).getNombre();
380     if(turno == 1) jugador = ejercitos.get(1).getNombre();
381     System.out.println("Ejercito Actual: " + jugador);
382
383     while(true){ //Verifica que los datos ingresados sean validos, tanto el ejercito como
384                 la direccion elegida
385         posicion = sc.nextLine();
386         System.out.println("Ingrese la direccion de movimiento ( w: arriba; s: abajo; a:
387                             izquierda; d: derecha)");
388         direccion = sc.nextLine();
389         if(CumpleCondicion2(posicion, direccion, ejercitos, jugador)) break;
390     }
391 }
392
393 public static void Jugar2(ArrayList<Ejercito> ejercitos, Reino reino){
394     CrearSoldados(ejercitos);
395     ImprimirTableroSoldados(ejercitos);
396     int turno = 0;
397     while(true){
```

```
386     Movimiento2(ejercitos, turno);
387     turno++;
388     ImprimirTableroSoldados(ejercitos);
389     System.out.println(ejercitos.get(0).getSoldados().size() + "x" +
        ejercitos.get(1).getSoldados().size());
390     if(ejercitos.get(0).getSoldados().size() == 0){
391         System.out.println("Se termino el enfrentamiento. Ganador: " +
            ejercitos.get(1).getNombre());
392         reino.getEjercitos().remove(reino.getEjercitos().indexOf(ejercitos.get(0)));
393         break;
394     }
395     if(ejercitos.get(1).getSoldados().size() == 0){
396         System.out.println("Se termino el enfrentamiento. Ganador: " +
            ejercitos.get(0).getNombre());
397         reino.getEjercitos().remove(reino.getEjercitos().indexOf(ejercitos.get(1)));
398         break;
399     }
400 }
401 }
402 }
```

- **Commit 2:** Se modificaron metodos del constructor Soldado

Listing 2: Clase Soldado

```
1  import java.util.*;
2
3  public class Soldado{
4      private String nombre;
5      private int nivelVida;
6      private int vidaActual;
7      private boolean vive;
8      private int ejercito;
9      private int fila;
10     private int columna;
11     private int nivelAtaque;
12     private int nivelDefensa;
13     private int velocidad;
14     private String actitud;
15
16     public Soldado(){
17         nivelVida = 0;
18         vidaActual = 0;
19         nivelAtaque = 0;
20         nivelDefensa = 0;
21         vive = true;
22     }
23
24     public Soldado(String nombre){
25         this.nombre = nombre;
26         nivelVida = 0;
27         vidaActual = 0;
28         nivelAtaque = 0;
29         nivelDefensa = 0;
30         vive = true;
    }
```

```
31     }
32
33     public Soldado(String nombre, int nivelVida, int ejercito, int fila, int columna){
34         this.nombre = nombre;
35         this.nivelVida = nivelVida;
36         this.vidaActual = nivelVida;
37         this.ejercito = ejercito;
38         this.fila = fila;
39         this.columna = columna;
40         vive = true;
41     }
42
43     public Soldado(String nombre, int nivelVida, int ejercito, int fila, int columna, int
44         nivelAtaque, int nivelDefensa, int velocidad){
45         this.nombre = nombre;
46         this.nivelVida = nivelVida;
47         this.vidaActual = nivelVida;
48         this.ejercito = ejercito;
49         this.fila = fila;
50         this.columna = columna;
51         this.nivelAtaque = nivelAtaque;
52         this.nivelDefensa = nivelDefensa;
53         this.velocidad = velocidad;
54         vive = true;
55     }
56
57     public void atacar(){
58         avanzar();
59         actitud = "ofensiva";
60     }
61
62     public void defender(){
63         velocidad = 0;
64         actitud = "defensiva";
65     }
66
67     public void avanzar(){
68         velocidad++;
69     }
70
71     public void retroceder(){
72         if(velocidad > 0){
73             velocidad = 0;
74             actitud = "defensiva";
75         } else velocidad --;
76     }
77
78     public void serAtacado(int num){
79         vidaActual -= num;
80     }
81
82     public void huir(){
83         velocidad += 2;
84         actitud = "fuga";
85     }
```

```
86 public void morir(){
87     vidaActual = 0;
88     vive = false;
89 }
90
91 public void cambiarPosicion(int vidaActual, int fila, int columna){
92     this.vidaActual = nivelVida;
93     this.fila = fila;
94     this.columna = columna;
95 }
96
97 public void vivir(){
98     vive = true;
99 }
100
101 public void setNombre(String nombre){
102     this.nombre = nombre;
103 }
104
105 public void setVidaActual(int vidaActual){
106     this.vidaActual = vidaActual;
107 }
108
109 public void setEjercito(int ejercito){
110     this.ejercito = ejercito;
111 }
112
113 public void setFila(int fila){
114     this.fila = fila;
115 }
116
117 public void setColumna(int columna){
118     this.columna = columna;
119 }
120
121 public void setAtaque(int nivelAtaque){
122     this.nivelAtaque = nivelAtaque;
123 }
124
125 public void setDefensa(int nivelDefensa){
126     this.nivelDefensa = nivelDefensa;
127 }
128
129 public void setVelocidad(int velocidad){
130     this.velocidad = velocidad;
131 }
132
133 public String getNombre(){
134     return nombre;
135 }
136
137 public int getVidaActual(){
138     return vidaActual;
139 }
140
141 public int getFila(){
```

```
142     return fila;
143 }
144
145 public int getEjercito(){
146     return ejercito;
147 }
148
149 public int getColumna(){
150     return columna;
151 }
152
153 public int getAtaque(){
154     return nivelAtaque;
155 }
156
157 public int getDefensa(){
158     return nivelDefensa;
159 }
160
161 public int getVelocidad(){
162     return velocidad;
163 }
164
165 public boolean getVive(){
166     return vive;
167 }
168 }
```

- **Commit 3:** Se creo la clase Reino para poder almacenar a los ejercitos, en los cuales se encuentran los soldados

Listing 3: Clase Reino

```
1 import java.util.*;
2
3 public class Reino {
4     private String reino;
5     private String letra;
6     private ArrayList<Ejercito> ejercitos = new ArrayList<>();
7
8     public Reino(String reino){
9         this.reino = reino;
10        this.letra = reino.substring(0,1);
11    }
12
13    public String getNombre(){
14        return reino;
15    }
16
17    public String getLetra(){
18        return letra;
19    }
20
21    public ArrayList<Ejercito> getEjercitos(){
22        return ejercitos;
23    }
24 }
```

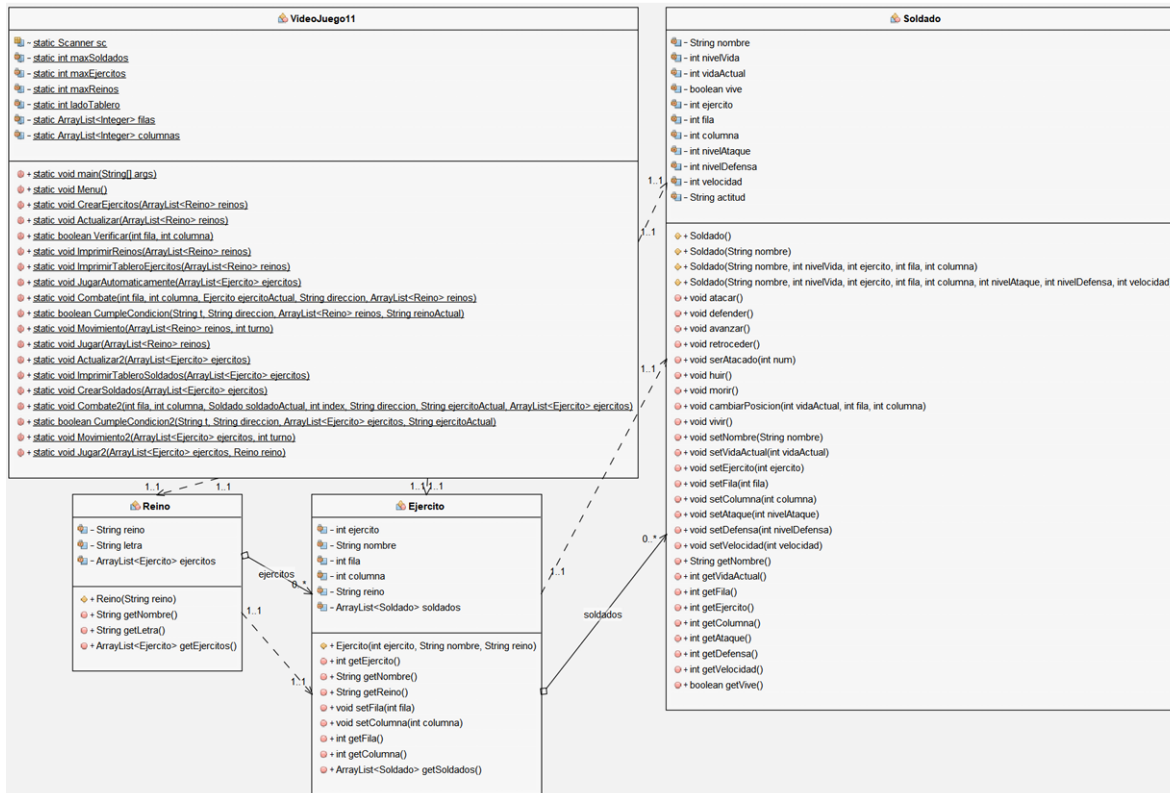
```
23 }  
24 }
```

- **Commit 4:** Se creo la clase Ejercito en el cual estarian almacenados todos los soldados

Listing 4: Clase Ejercito

```
1 import java.util.*;  
2  
3 public class Ejercito {  
4     private int ejercito;  
5     private String nombre;  
6     private int fila;  
7     private int columna;  
8     private String reino;  
9     private ArrayList<Soldado> soldados = new ArrayList<>();  
10  
11     public Ejercito(int ejercito, String nombre, String reino){  
12         this.ejercito = ejercito;  
13         this.nombre = nombre;  
14         this.reino = reino;  
15     }  
16  
17     public int getEjercito(){  
18         return ejercito;  
19     }  
20  
21     public String getNombre(){  
22         return nombre;  
23     }  
24  
25     public String getReino(){  
26         return reino;  
27     }  
28  
29     public void setFila(int fila){  
30         this.fila = fila;  
31     }  
32  
33     public void setColumna(int columna){  
34         this.columna = columna;  
35     }  
36  
37     public int getFila(){  
38         return fila;  
39     }  
40  
41     public int getColumna(){  
42         return columna;  
43     }  
44  
45     public ArrayList<Soldado> getSoldados(){  
46         return soldados;  
47     }  
48 }
```

- **Commit 5 :** El diagrama UML no da a conocer todos los aspectos del programa



## 5. Programa Compilado

**Commit 6:** Todas las funciones del Juego funcionan correctamente

```

----- SOLDADOS EN TODOS LOS REINOS -----
Reino: Inglaterra(I)
- Ejército: I1 Posicion: 8x9
- Ejército: I2 Posicion: 4x9
Reino: Moros(M)
- Ejército: M1 Posicion: 6x1
- Ejército: M2 Posicion: 3x3
- Ejército: M3 Posicion: 5x4
- Ejército: M4 Posicion: 10x5

+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | M | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | I | |
+---+---+---+---+---+---+---+---+
| | | | M | | | | | |
+---+---+---+---+---+---+---+---+
| M | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | I | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | M | | | | | |
+---+---+---+---+---+---+---+---+

Jugador 1: Inglaterra Jugador2: Moros
Ingrese la posicion del ejercito a mover. Ejemplo (1x2)

```

```

Reino Actual: Inglaterra
8x9
Ingrese la direccion de movimiento ( w: arriba; s: abajo; a: izquierda; d: derecha)
s

+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | M | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | I | |
+---+---+---+---+---+---+---+---+
| | | | M | | | | | |
+---+---+---+---+---+---+---+---+
| M | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | I | |
+---+---+---+---+---+---+---+---+
| | | | M | | | | | |
+---+---+---+---+---+---+---+---+

Jugador 1: Inglaterra Jugador2: Moros
Ingrese la posicion del ejercito a mover. Ejemplo (1x2)

```



```

Reino Actual: Moros
10x5
Ingrese la direccion de movimiento ( w: arriba; s: abajo; a: izquierda; d: derecha)
w

+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | M | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | I | |
+---+---+---+---+---+---+---+---+
| | | | M | | | | | | |
+---+---+---+---+---+---+---+---+
| M | | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | M | | | | I | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | | |
+---+---+---+---+---+---+---+---+

Jugador 1: Inglaterra Jugador2: Moros
Ingrese la posicion del ejercito a mover. Ejemplo (1x2)
    
```

```

Reino Actual: Inglaterra
9x9
Ingrese la direccion de movimiento ( w: arriba; s: abajo; a: izquierda; d: derecha)
a

+---+---+---+---+---+---+---+---+---+
| | | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | M | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | I | |
+---+---+---+---+---+---+---+---+
| | | | M | | | | | | |
+---+---+---+---+---+---+---+---+
| M | | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | M | | | I | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | | |
+---+---+---+---+---+---+---+---+

Jugador 1: Inglaterra Jugador2: Moros
Ingrese la posicion del ejercito a mover. Ejemplo (1x2)
    
```

```

Reino Actual: Moros
9x5
Ingrese la direccion de movimiento ( w: arriba; s: abajo; a: izquierda; d: derecha)
d

+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | M | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | I | |
+---+---+---+---+---+---+---+---+
| | | M | | | | | |
+---+---+---+---+---+---+---+---+
| M | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | M | I | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+

Jugador 1: Inglaterra Jugador2: Moros
Ingrese la posicion del ejercito a mover. Ejemplo (1x2)
1 2

Reino Actual: Inglaterra
9x8
Ingrese la direccion de movimiento ( w: arriba; s: abajo; a: izquierda; d: derecha)
a

+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | M | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | I | |
+---+---+---+---+---+---+---+---+
| | | M | | | | | |
+---+---+---+---+---+---+---+---+
| M | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | M | I | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+

Jugador 1: Inglaterra Jugador2: Moros
Ingrese la posicion del ejercito a mover. Ejemplo (1x2)
1 2

Reino Actual: Moros
9x6
Ingrese la direccion de movimiento ( w: arriba; s: abajo; a: izquierda; d: derecha)
d
    
```

```
***** QUE DESEA REALIZAR *****
1.Jugar Manualmente    2.Jugar Automaticamente
1

+---+---+---+---+---+---+---+---+
| | | | | | | | 2 | |
+---+---+---+---+---+---+---+---+
| | | | 2 | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | 2 | |
+---+---+---+---+---+---+---+---+
| | 2 | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | 1 | |
+---+---+---+---+---+---+---+---+
| | | | 1 | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | 2 | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+

Jugador 1: Ejercito I1 Jugador2: Ejercito M4
Ingrese la posicion del soldado a mover. Ejemplo (1x2)
Ejercito Actual: I1
8x6
Ingrese la direccion de movimiento ( w: arriba; s: abajo; a: izquierda; d: derecha)
s
Soldado no encontrado, por favor ingrese los datos de nuevo
8x6
Ingrese la direccion de movimiento ( w: arriba; s: abajo; a: izquierda; d: derecha)
w
Soldado no encontrado, por favor ingrese los datos de nuevo
7x5
Ingrese la direccion de movimiento ( w: arriba; s: abajo; a: izquierda; d: derecha)
d

+---+---+---+---+---+---+---+---+
| | | | | | | | 2 | |
+---+---+---+---+---+---+---+---+
| | | | 2 | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | 2 | |
+---+---+---+---+---+---+---+---+
| | 2 | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | 1 | |
+---+---+---+---+---+---+---+---+
| | | | | 1 | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | 2 | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+

2x5
Jugador 1: Ejercito I1 Jugador2: Ejercito M4
Ingrese la posicion del soldado a mover. Ejemplo (1x2)
Ejercito Actual: M4
4x9
Ingrese la direccion de movimiento ( w: arriba; s: abajo; a: izquierda; d: derecha)
s
```

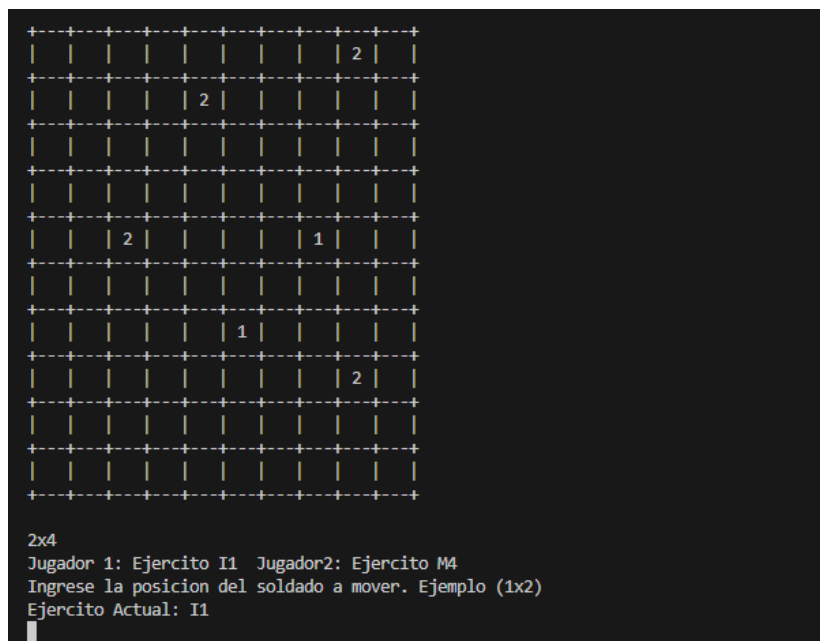
```

+---+---+---+---+---+---+---+---+
| | | | | | | | 2 | |
+---+---+---+---+---+---+---+---+
| | | | 2 | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | 2 | | | | | 2 | |
+---+---+---+---+---+---+---+---+
| | | | | | | 1 | |
+---+---+---+---+---+---+---+---+
| | | | | 1 | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | 2 | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+

2x5
Jugador 1: Ejercito I1 Jugador2: Ejercito M4
Ingrese la posicion del soldado a mover. Ejemplo (1x2)
Ejercito Actual: I1
6x8
Ingrese la direccion de movimiento ( w: arriba; s: abajo; a: izquierda; d: derecha)
w

+---+---+---+---+---+---+---+---+
| | | | | | | | 2 | |
+---+---+---+---+---+---+---+---+
| | | | 2 | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | 2 | | | | | 1 | 2 | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | 1 | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | 2 | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+
| | | | | | | | | |
+---+---+---+---+---+---+---+---+

2x5
Jugador 1: Ejercito I1 Jugador2: Ejercito M4
Ingrese la posicion del soldado a mover. Ejemplo (1x2)
Ejercito Actual: M4
5x9
Ingrese la direccion de movimiento ( w: arriba; s: abajo; a: izquierda; d: derecha)
a
Probabilidades de ganar:
Soldado actual: 50.0% --- Soldado enemigo: 50.0%
El Soldado enemigo gana
    
```



## 6. Referencias

- [https://www.w3schools.com/js/js\\_ajax\\_intro.asp](https://www.w3schools.com/js/js_ajax_intro.asp)
- <https://www.youtube.com/watch?v=cAqmF7mtZv0>
- <https://www.youtube.com/watch?v=cAqmF7mtZv0>