

Universidad Nacional de San Agustín
Escuela Profesional de Ingeniería de Sistemas
Fundamentos de Programación II
Práctica de Laboratorio N° 15:
Definición de Clases de Usuario
Clase Ejército – Soldado – Mapa

Nombre: Jhonatan Benjamin Mamani Céspedes

CUI: 20232188

Link de GitHub: <https://github.com/JBenjamin01/fp2-24b/tree/main/Laboratorio>

1. Basándose en la clase Soldado de las prácticas anteriores.
2. Crear la clase Mapa, que esté constituida por el tablero antes visto, pero que en lugar de soldados, posicione ejércitos en ciertas posiciones aleatorias (entre 1 y 10 ejércitos por cada reino). Se deben generar ejércitos de 2 reinos, entre 1 y 10 soldados por ejército. No se admite guerra civil. El Mapa tiene como atributo el tipo de territorio que puede ser (bosque, campo abierto, montaña, desierto, playa). La cantidad de ejércitos y cantidad de soldados por ejército, así como todos sus atributos se deben generar aleatoriamente
3. El mapa tiene un solo tipo de territorio autogenerado
4. Considerar que el territorio influye en los resultados de las batallas, así cada reino tiene bonus según el territorio: Inglaterra->bosque, Francia->campo abierto, Castilla-Aragón->montaña, Moros->desierto, Sacro Imperio Romano Germánico->bosque, playa y campo abierto. En dichos casos, se aumenta el nivel de vida en 1 a todos los soldados del reino beneficiado
5. Dibujar el Mapa con las restricciones que sólo 1 ejercito como máximo en cada cuadrado
6. Realizar el diagrama de clases de UML completo
7. Indicar quién ganaría la guerra y justificar por qué. Describir 3 métricas usadas
8. Hacerlo iterativo



Clase Soldado.java:

```
1 public class Soldado {
2     public static final int MAX_SOLDADOS_POR_EJERCITO = 10;
3     public static int totalSoldadosCreados = 0;
4     public static int soldadosEjercito1 = 0;
5     public static int soldadosEjercito2 = 0;
6     private String nombre;
7     private int puntosVida;
8     private int fila;
9     private char columna;
10    private int nivelAtaque;
11    private int nivelDefensa;
12    private int velocidad;
13    private String actitud;
14    private boolean vive;
15    private int ejercito;
16
17    // Constructores sobrecargados del Soldado
18    public Soldado(String nombre, int puntosVida, int fila, char columna, int nivelAtaque, int nivelDefensa, int ejercito) {
19        this.nombre = nombre;
20        this.puntosVida = puntosVida;
21        this.fila = fila;
22        this.columna = columna;
23        this.nivelAtaque = nivelAtaque;
24        this.nivelDefensa = nivelDefensa;
25        this.velocidad = 0;
26        this.actitud = "Defensiva";
27        this.vive = true;
28        this.ejercito = ejercito;
29        totalSoldadosCreados++;
30    }
31    public Soldado(String nombre, int puntosVida, int fila, char columna, int ejercito) {
32        this.nombre = nombre;
33        this.puntosVida = puntosVida;
34        this.fila = fila;
35        this.columna = columna;
36        this.nivelAtaque = (int)(Math.random() * 5) + 1;
37        this.nivelDefensa = (int)(Math.random() * 5) + 1;
38        this.velocidad = 0;
39        this.actitud = "Defensiva";
40        this.vive = true;
41        this.ejercito = ejercito;
42        totalSoldadosCreados++;
43    }
44    public Soldado(String nombre, int nivelVida, int ejercito) {
45        this.nombre = nombre;
46        this.nivelVida = nivelVida;
47        this.fila = 0;
48        this.columna = 'A';
49        this.nivelAtaque = (int)(Math.random() * 5) + 1;
50        this.nivelDefensa = (int)(Math.random() * 5) + 1;
51        this.velocidad = 0;
52        this.actitud = "Defensiva";
53        this.vive = true;
54        this.ejercito = ejercito;
55        totalSoldadosCreados++;
56    }
57
58    // Getters
59    public String getNombre() {
60        return nombre;
61    }
62    public int getFila() {
63        return fila;
64    }
65    public char getColumna() {
66        return columna;
67    }
68    public int getNivelVida() {
69        return nivelVida;
70    }
71    public int getNivelAtaque() {
72        return nivelAtaque;
73    }
74    public int getNivelDefensa() {
75        return nivelDefensa;
76    }
77    public int getVelocidad() {
78        return velocidad;
79    }
```

```

79     }
80     public String getActitud() {
81         return actitud;
82     }
83     public boolean isVivo() {
84         return vive;
85     }
86     public int getEjercito() {
87         return ejercito;
88     }
89
90     // Setters
91     public void setFila(int fila) {
92         this.fila = fila;
93     }
94     public void setColumna(char columna) {
95         this.columna = columna;
96     }
97     public void setNivelVida(int nivelVida) {
98         this.nivelVida = nivelVida;
99     }
100    public void setNivelAtaque(int nivelAtaque) {
101        this.nivelAtaque = nivelAtaque;
102    }
103    public void setNivelDefensa(int nivelDefensa) {
104        this.nivelDefensa = nivelDefensa;
105    }
106    public void setEjercito(int ejercito) {
107        this.ejercito = ejercito;
108    }
109
110    // Métodos del UML de referencia
111    public void atacar() {
112        velocidad += 1;
113        actitud = "Ofensiva";
114        System.out.println(nombre + " ha atacado, su velocidad es ahora " + velocidad);
115    }
116    public void defender() {
117        actitud = "Defensiva";
118        System.out.println(nombre + " está en modo defensivo.");
119    }
120    public void huir() {
121        velocidad += 2;
122        actitud = "Fuga";
123        System.out.println(nombre + " está huyendo, su velocidad es ahora " + velocidad);
124    }
125    public void avanzar() {
126        velocidad += 1;
127        System.out.println(nombre + " avanza, su velocidad es ahora " + velocidad);
128    }
129    public void retroceder() {
130        if (velocidad > 0) {
131            velocidad = 0;
132            actitud = "Defensiva";
133            System.out.println(nombre + " se ha detenido, velocidad actual: " + velocidad);
134        } else {
135            velocidad -= 1;
136            System.out.println(nombre + " ha retrocedido, velocidad negativa: " + velocidad);
137        }
138    }
139    public void serAtacado(int daño) {
140        recibirAtaque(daño);
141    }
142    public void recibirAtaque(int daño) {
143        nivelVida -= daño;
144        if (nivelVida <= 0) {
145            nivelVida = 0;
146            System.out.println(nombre + " ha muerto.");
147        } else {
148            System.out.println(nombre + " ha recibido " + daño + " de daño. Vida restante: " + nivelVida);
149        }
150    }
151    public void morir() {
152        vive = false;
153        System.out.println(nombre + " ha muerto.");
154    }
155
156    @Override
157    public String toString() {
158        return "Nombre: " + nombre
159            + " | Vida: " + nivelVida
160            + " | Ataque: " + nivelAtaque
161            + " | Defensa: " + nivelDefensa

```

```

161         + " | Defensa: " + nivelDefensa
162         + " | Velocidad: " + velocidad
163         + " | Actitud: " + actitud
164         + " | Posición: " + columna + fila
165         + " | Vive: " + (vive ? "Si" : "No");
166     }
167 }

```

Clase Ejercito.java

```

1  import java.util.*;
2  public class Ejercito {
3      private int numEjercito;
4      private String reino;
5      private ArrayList<Soldado> soldados = new ArrayList<Soldado>();
6      private int fila;
7      private char columna;
8
9      public Ejercito(String reino, int numEjercito, int fila, char columna) {
10         this.numEjercito = numEjercito;
11         this.reino = reino;
12         this.fila = fila;
13         this.columna = columna;
14         generarSoldados();
15     }
16
17     public void addSoldado(Soldado soldado) {
18         soldados.add(soldado);
19     }
20
21     public int getTotalSoldadosEjercito() {
22         return soldados.size();
23     }
24
25     public int getTotalVidaEjercito() {
26         int total = 0;
27         for (Soldado s : soldados) {
28             total += s.getNivelVida();
29         }
30         return total;
31     }
32
33     public String getReino() {
34         return reino;
35     }
36
37     public int getFila() {
38         return fila;
39     }
40
41     public char getColumna() {
42         return columna;
43     }
44
45     public int getCantidadSoldados() {
46         return soldados.size();
47     }
48
49     public int getVidaTotal() {
50         int vidaTotal = 0;
51         for (Soldado s : soldados) {
52             vidaTotal += s.getNivelVida();
53         }
54         return vidaTotal;
55     }
56
57     private void generarSoldados() {
58         Set<String> posicionesOcupadas = new HashSet<>();
59         int n = (int) (Math.random() * Soldado.MAX_SOLDADOS_POR_EJERCITO) + 1;
60         for (int i = 0; i < n; i++) {
61             String nombre = "Soldado" + numEjercito + "X" + (i + 1);
62             int puntosVida = (int) (Math.random() * 5) + 1;
63             int puntosAtaque = (int) (Math.random() * 5) + 1;
64             int puntosDefensa = (int) (Math.random() * 5) + 1;
65
66             int fila, columna;
67

```

```

68         do {
69             fila = (int) (Math.random() * 10);
70             columna = (int) (Math.random() * 10);
71         } while (posicionesOcupadas.contains(fila + "," + columna));
72
73         posicionesOcupadas.add(fila + "," + columna);
74
75         Soldado soldado = new Soldado(nombre, puntosVida, 1 + fila, (char) ('A' + columna), puntosAtaque, puntosDefensa, i + 1);
76
77         soldados.add(soldado);
78     }
79 }
80
81 public void mostrarEjercito() {
82     System.out.println("\nSoldados de este ejército:");
83     for (Soldado soldado : soldados) {
84         System.out.println(" - " + soldado);
85     }
86     System.out.println();
87 }
88
89 public void aplicarBonus(String tipoTerritorio) {
90     if (tipoTerritorio.equals("bosque") && (reino.equals("Inglaterra") || reino.equals("Sacro Imperio Romano Germánico"))) {
91         for (Soldado s : soldados) {
92             s.setNivelVida(s.getNivelVida() + 1);
93         }
94     } else if (tipoTerritorio.equals("campo abierto") && reino.equals("Francia")) {
95         for (Soldado s : soldados) {
96             s.setNivelVida(s.getNivelVida() + 1);
97         }
98     } else if (tipoTerritorio.equals("montaña") && (reino.equals("Castilla-Aragón"))) {
99         for (Soldado s : soldados) {
100             s.setNivelVida(s.getNivelVida() + 1);
101         }
102     } else if (tipoTerritorio.equals("desierto") && reino.equals("Moros")) {
103         for (Soldado s : soldados) {
104             s.setNivelVida(s.getNivelVida() + 1);
105         }
106     } else if (tipoTerritorio.equals("playa") && (reino.equals("Sacro Imperio Romano Germánico") || reino.equals("Francia"))) {
107         for (Soldado s : soldados) {
108             s.setNivelVida(s.getNivelVida() + 1);
109         }
110     }
111 }
112 }

```

Clase Reino.java

```

1  import java.util.*;
2  public class Reino {
3      private static ArrayList<String> reinos = new ArrayList<String>(Arrays.asList("Inglaterra", "Francia", "Castilla-Aragón",
4                                          "Moros", "Sacro Imperio Romano Germánico"));
5
6      private String nombre;
7      private ArrayList<Ejercito> ejercitos = new ArrayList<>();
8      private static Set<String> posicionesOcupadas = new HashSet<>();
9
10     public Reino() {
11         generarNombre();
12         generarEjercitos();
13     }
14
15     private void generarNombre() {
16         Random r = new Random();
17         int indice = r.nextInt(reinos.size());
18         nombre = reinos.remove(indice);
19     }
20
21     public String getNombre() {
22         return nombre;
23     }
24
25     public ArrayList<Ejercito> getEjercitos() {
26         return ejercitos;
27     }
28
29     public void setNombre(String nombre) {
30         this.nombre = nombre;
31     }
32
33     public void setEjercitos(ArrayList<Ejercito> ejercitos) {
34         this.ejercitos = ejercitos;
35     }
36
37     public void setPosicionesOcupadas(Set<String> posicionesOcupadas) {
38         this.posicionesOcupadas = posicionesOcupadas;
39     }
40 }

```

```

28     public int getTotalSoldadosReino() {
29         int total = 0;
30         for (Ejercito e : ejercitos) {
31             total += e.getTotalSoldadosEjercito();
32         }
33         return total;
34     }
35
36     public int getTotalVidaReino() {
37         int total = 0;
38         for (Ejercito e : ejercitos) {
39             total += e.getTotalVidaEjercito();
40         }
41         return total;
42     }
43
44     private void generarEjercitos() {
45         int n = (int) (Math.random() * 10) + 1;
46         for (int i = 0; i < n; i++) {
47             int fila, columna;
48             do {
49                 fila = (int) (Math.random() * 10);
50                 columna = (int) (Math.random() * 10);
51             } while (posicionesOcupadas.contains(fila + "," + columna));
52
53             posicionesOcupadas.add(fila + "," + columna);
54             Ejercito e = new Ejercito(nombre, i + 1, 1 + fila, (char) ('A' + columna));
55
56             ejercitos.add(e);
57         }
58     }
59
60     public boolean contains(Ejercito ejercito) {
61         return ejercitos.contains(ejercito);
62     }
63
64     public void mostrarInformacion() {
65         System.out.println("■ REINO DE " + nombre.toUpperCase() + " ■");
66         System.out.println("+ Total de ejércitos: " + ejercitos.size());
67         System.out.println("+ Total de soldados: " + getTotalSoldadosReino());
68         System.out.println("+ Total de vida: " + getTotalVidaReino());
69         System.out.println("\nLista de ejércitos creados para este reino:\n");
70         for (Ejercito ejercito : ejercitos) {
71             System.out.println("\nEjercito " + (ejercitos.indexOf(ejercito) + 1) + " ---> "
72                 + ejercito.getCantidadSoldados() + " soldados, "
73                 + ejercito.getTotalVidaEjercito() + " puntos de vida, ubicado en la posición "
74                 + ejercito.getColuma() + ejercito.getFila());
75             ejercito.mostrarEjercito();
76         }
77     }
78 }
79

```

Clase Mapa.java

```

1  import java.util.*;
2
3  public class Mapa {
4      private static final String [] territorios = {"bosque", "campo abierto", "montaña", "desierto", "playa"};
5
6      private String territorio;
7      private ArrayList<ArrayList<Ejercito>> tablero = new ArrayList<>();
8      private Reino r1;
9      private Reino r2;
10
11     public Mapa(Reino r1, Reino r2) {
12         this.r1 = r1;
13         this.r2 = r2;
14         generarTerritorio();
15         inicializarTablero();
16         colocarEjercitos();
17     }
18
19     // Generar un tipo de territorio aleatorio para el mapa
20     private void generarTerritorio() {
21         Random rand = new Random();
22         String tipoTerreno = territorios[rand.nextInt(territorios.length)];
23         territorio = tipoTerreno;
24     }
25
26     public String getTerritorio() {
27         return territorio;
28     }
29

```

```

30 public void inicializarTablero() {
31     for (int i = 0; i < 10; i++) {
32         ArrayList<Ejercito> fila = new ArrayList<>();
33         for (int j = 0; j < 10; j++)
34             fila.add(null);
35         tablero.add(fila);
36     }
37 }
38
39 private void colocarEjercitos() {
40     for (Ejercito e : r1.getEjercitos()) {
41         int fila = e.getFila() - 1;
42         int columna = e.getColumna() - 'A';
43
44         tablero.get(fila).set(columna, e);
45     }
46
47     for (Ejercito e : r2.getEjercitos()) {
48         int fila = e.getFila() - 1;
49         int columna = e.getColumna() - 'A';
50
51         tablero.get(fila).set(columna, e);
52     }
53 }
54
55 public void mostrarMapa() {
56     System.out.println("\n=====");
57     System.out.println("          CAMPO DE BATALLA          ");
58     System.out.println("=====");
59     System.out.println("\n ESCENARIO: " + territorio.toUpperCase() + " \n");
60     System.out.println("    * Los ejércitos del reino de " + r1.getNombre() + " están colocados entre corchetes ([x])");
61     System.out.println("    * Los del reino de " + r2.getNombre() + " entre signos menor y mayor que (<x>)");
62     System.out.println("=====");
63
64     System.out.println("\n          A          B          C          D          E          F          G          H          I          J");
65     System.out.println();
66     System.out.println("-----");
67
68     for (int i = 0; i < tablero.size(); i++) {
69         System.out.print("    " + (i + 1) + "\t| ");
70         for (int j = 0; j < tablero.get(i).size(); j++) {
71             Ejercito ejercito = tablero.get(i).get(j);
72             if (ejercito == null)
73                 System.out.print("          | ");
74             else {
75                 int totalSoldados = ejercito.getTotalSoldadosEjercito();
76                 int vidaTotal = ejercito.getVidaTotal();
77
78                 String soldados = (totalSoldados < 10 ? " " : "") + totalSoldados;
79                 String vida = vidaTotal + (vidaTotal < 10 ? " " : "");
80
81                 if (r1.contains(ejercito)) {
82                     System.out.print("[ " + soldados + "-" + vida + " ] | ");
83                 } else if (r2.contains(ejercito)) {
84                     System.out.print("< " + soldados + "-" + vida + " > | ");
85                 } else {
86                     System.out.print(totalSoldados + "   | ");
87                 }
88             }
89         }
90         System.out.println();
91         System.out.println("-----");
92     }
93     System.out.println("\n===== \n");
94 }
95 }

```

Clase Videojuego.java

```
1 // LABORATORIO N° 15
2 // AUTOR: JHONATAN BENJAMIN MAMANI CÉSPEDES
3 // TIEMPO: 167 MINUTOS
4 import java.util.*;
5 public class VideoJuego {
6     public static void main(String[] args) {
7         // Los reinos que creo aquí aleatoriamente generan sus ejércitos y soldados
8         // desde sus respectivas clases:
9         Reino r1 = new Reino();
10        Reino r2 = new Reino();
11
12        Mapa mapa = new Mapa(r1, r2);
13        mapa.mostrarMapa();
14
15        System.out.println("Información de los reinos para comprobar los datos de ejércitos y soldados creados:\n");
16        r1.mostrarInformacion();
17        System.out.println();
18        r2.mostrarInformacion();
19        System.out.println("=====
20        + "===== \n");
21
22        // Aquí estan las métricas que definirán a un vencedor de Los reinos
23        System.out.println("Las 3 métricas usadas para determinar al vencedor:\n");
24        mostrarMetricasVencedor(r1, r2);
25    }
26
27    public static void mostrarMetricasVencedor(Reino r1, Reino r2) {
28        // Para definir al vencedor voy a utilizar estas tres métricas diferentes;
29        // Total de soldados, total de niveles de vida y probabilidad respecto a la vida
30        System.out.println("1. Cantidad de soldados vivos");
31        System.out.println("  Cantidad de soldados vivos del reino de " + r1.getNombre() + ": " + r1.getTotalSoldadosReino());
32        System.out.println("  Cantidad de soldados vivos del reino de " + r2.getNombre() + ": " + r2.getTotalSoldadosReino());
33        System.out.println("  .....");
34        if (r1.getTotalSoldadosReino() > r2.getTotalSoldadosReino()) {
35            System.out.println("    El reino de " + r1.getNombre() + " ha ganado la batalla.");
36        } else if (r1.getTotalSoldadosReino() < r2.getTotalSoldadosReino()) {
37            System.out.println("    El reino de " + r2.getNombre() + " ha ganado la batalla.");
38        } else {
39            System.out.println("    La batalla ha terminado en empate.");
40        }
41
42        System.out.println();
43
44        System.out.println("2. Total de niveles de vida");
45        System.out.println("  Total de niveles de vida del reino de " + r1.getNombre() + ": " + r1.getTotalVidaReino());
46        System.out.println("  Total de niveles de vida del reino de " + r2.getNombre() + ": " + r2.getTotalVidaReino());
47        System.out.println("  .....");
48        if (r1.getTotalVidaReino() > r2.getTotalVidaReino()) {
49            System.out.println("    El reino de " + r1.getNombre() + " ha ganado la batalla.");
50        } else if (r1.getTotalVidaReino() < r2.getTotalVidaReino()) {
51            System.out.println("    El reino de " + r2.getNombre() + " ha ganado la batalla.");
52        } else {
53            System.out.println("    La batalla ha terminado en empate.");
54        }
55
56        System.out.println();
57
58        System.out.println("3. Probabilidades respecto a sus puntos de vida");
59        int totalVida = r1.getTotalVidaReino() + r2.getTotalVidaReino();
60        double probabilidadR1 = (double) r1.getTotalVidaReino() / totalVida;
61        double probabilidadR2 = (double) r2.getTotalVidaReino() / totalVida;
62        System.out.println("  Probabilidad de vida total del reino de " + r1.getNombre() + ": " + probabilidadR1 * 100 + "%");
63        System.out.println("  Probabilidad de vida roal del reino de " + r2.getNombre() + ": " + probabilidadR2 * 100 + "%");
64        System.out.println("\n    Calculando el resultado de la batalla...");
65        System.out.println();
66        Random r = new Random();
67        double resultado = r.nextDouble() * 100;
68        if (resultado < probabilidadR1 * 100) {
69            System.out.println("    El reino de " + r1.getNombre() + " ha ganado la batalla.");
70        } else if (resultado > probabilidadR1 * 100) {
71            System.out.println("    El reino de " + r2.getNombre() + " ha ganado la batalla.");
72        } else {
73            System.out.println("    La batalla ha terminado en empate.");
74        }
75        System.out.println("\n=====
76        + "===== \n");
77        System.out.println();
78    }
79 }
```


Consola:

CAMPO DE BATALLA

ESCENARIO: BOSQUE

* Los ejércitos del reino de Inglaterra están colocados entre corchetes ([x])
 * Los del reino de Francia entre signos menor y mayor que (<x>)

	A	B	C	D	E	F	G	H	I	J
1										[5-18]
2				[8-15]						
3			< 8-22>							
4								[6-16]		
5			< 4-17>							
6	[10-33]			< 4-9 >			[10-31]			
7		[6-18]						< 2-6 > < 5-17>		
8							[8-24]			
9										[7-19]
10										

Información de los reinos para comprobar los datos de ejércitos y soldados creados:

REINO DE INGLATERRA

+ Total de ejércitos: 8
 + Total de soldados: 60
 + Total de vida: 174

Lista de ejércitos creados para este reino:

Ejercito 1 ---> 5 soldados, 18 puntos de vida, ubicado en la posición J1

Soldados de este ejército:

- Nombre: Soldado1X1 | Vida: 4 | Ataque: 4 | Defensa: 1 | Velocidad: 0 | Actitud: Defensiva | Posición: A8 | Vive: Si
 - Nombre: Soldado1X2 | Vida: 4 | Ataque: 1 | Defensa: 1 | Velocidad: 0 | Actitud: Defensiva | Posición: E6 | Vive: Si
 - Nombre: Soldado1X3 | Vida: 2 | Ataque: 5 | Defensa: 2 | Velocidad: 0 | Actitud: Defensiva | Posición: G5 | Vive: Si
 - Nombre: Soldado1X4 | Vida: 4 | Ataque: 5 | Defensa: 4 | Velocidad: 0 | Actitud: Defensiva | Posición: G1 | Vive: Si
 - Nombre: Soldado1X5 | Vida: 4 | Ataque: 2 | Defensa: 2 | Velocidad: 0 | Actitud: Defensiva | Posición: B7 | Vive: Si

Ejercito 2 ---> 10 soldados, 33 puntos de vida, ubicado en la posición A6

Soldados de este ejército:

- Nombre: Soldado2X1 | Vida: 3 | Ataque: 4 | Defensa: 1 | Velocidad: 0 | Actitud: Defensiva | Posición: H3 | Vive: Si
 - Nombre: Soldado2X2 | Vida: 3 | Ataque: 5 | Defensa: 3 | Velocidad: 0 | Actitud: Defensiva | Posición: C6 | Vive: Si
 - Nombre: Soldado2X3 | Vida: 4 | Ataque: 5 | Defensa: 2 | Velocidad: 0 | Actitud: Defensiva | Posición: F1 | Vive: Si
 - Nombre: Soldado2X4 | Vida: 2 | Ataque: 4 | Defensa: 2 | Velocidad: 0 | Actitud: Defensiva | Posición: G4 | Vive: Si
 - Nombre: Soldado2X5 | Vida: 3 | Ataque: 4 | Defensa: 3 | Velocidad: 0 | Actitud: Defensiva | Posición: A4 | Vive: Si
 - Nombre: Soldado2X6 | Vida: 2 | Ataque: 4 | Defensa: 3 | Velocidad: 0 | Actitud: Defensiva | Posición: A10 | Vive: Si
 - Nombre: Soldado2X7 | Vida: 5 | Ataque: 4 | Defensa: 5 | Velocidad: 0 | Actitud: Defensiva | Posición: A3 | Vive: Si
 - Nombre: Soldado2X8 | Vida: 4 | Ataque: 5 | Defensa: 2 | Velocidad: 0 | Actitud: Defensiva | Posición: D6 | Vive: Si
 - Nombre: Soldado2X9 | Vida: 4 | Ataque: 5 | Defensa: 2 | Velocidad: 0 | Actitud: Defensiva | Posición: I2 | Vive: Si
 - Nombre: Soldado2X10 | Vida: 3 | Ataque: 1 | Defensa: 2 | Velocidad: 0 | Actitud: Defensiva | Posición: J4 | Vive: Si

Ejercito 3 ---> 8 soldados, 15 puntos de vida, ubicado en la posición D2

Soldados de este ejército:

- Nombre: Soldado3X1 | Vida: 5 | Ataque: 3 | Defensa: 5 | Velocidad: 0 | Actitud: Defensiva | Posición: G1 | Vive: Si
 - Nombre: Soldado3X2 | Vida: 1 | Ataque: 1 | Defensa: 1 | Velocidad: 0 | Actitud: Defensiva | Posición: C4 | Vive: Si
 - Nombre: Soldado3X3 | Vida: 1 | Ataque: 5 | Defensa: 1 | Velocidad: 0 | Actitud: Defensiva | Posición: B6 | Vive: Si
 - Nombre: Soldado3X4 | Vida: 1 | Ataque: 2 | Defensa: 3 | Velocidad: 0 | Actitud: Defensiva | Posición: B1 | Vive: Si
 - Nombre: Soldado3X5 | Vida: 2 | Ataque: 4 | Defensa: 4 | Velocidad: 0 | Actitud: Defensiva | Posición: I1 | Vive: Si
 - Nombre: Soldado3X6 | Vida: 1 | Ataque: 1 | Defensa: 2 | Velocidad: 0 | Actitud: Defensiva | Posición: I9 | Vive: Si
 - Nombre: Soldado3X7 | Vida: 3 | Ataque: 2 | Defensa: 4 | Velocidad: 0 | Actitud: Defensiva | Posición: E8 | Vive: Si
 - Nombre: Soldado3X8 | Vida: 1 | Ataque: 4 | Defensa: 2 | Velocidad: 0 | Actitud: Defensiva | Posición: G4 | Vive: Si

Ejercito 4 ---> 10 soldados, 31 puntos de vida, ubicado en la posición H6

Soldados de este ejército:

- Nombre:	Soldado4X1	Vida: 4	Ataque: 4	Defensa: 1	Velocidad: 0	Actitud: Defensiva	Posición: A2	Vive: Si
- Nombre:	Soldado4X2	Vida: 1	Ataque: 4	Defensa: 3	Velocidad: 0	Actitud: Defensiva	Posición: I1	Vive: Si
- Nombre:	Soldado4X3	Vida: 3	Ataque: 4	Defensa: 3	Velocidad: 0	Actitud: Defensiva	Posición: C5	Vive: Si
- Nombre:	Soldado4X4	Vida: 3	Ataque: 1	Defensa: 3	Velocidad: 0	Actitud: Defensiva	Posición: E5	Vive: Si
- Nombre:	Soldado4X5	Vida: 4	Ataque: 4	Defensa: 5	Velocidad: 0	Actitud: Defensiva	Posición: E3	Vive: Si
- Nombre:	Soldado4X6	Vida: 3	Ataque: 2	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: I8	Vive: Si
- Nombre:	Soldado4X7	Vida: 3	Ataque: 2	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: J4	Vive: Si
- Nombre:	Soldado4X8	Vida: 2	Ataque: 5	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: F10	Vive: Si
- Nombre:	Soldado4X9	Vida: 5	Ataque: 1	Defensa: 3	Velocidad: 0	Actitud: Defensiva	Posición: H1	Vive: Si
- Nombre:	Soldado4X10	Vida: 3	Ataque: 4	Defensa: 1	Velocidad: 0	Actitud: Defensiva	Posición: B7	Vive: Si

Ejercito 5 ---> 7 soldados, 19 puntos de vida, ubicado en la posición J9

Soldados de este ejército:

- Nombre:	Soldado5X1	Vida: 4	Ataque: 5	Defensa: 5	Velocidad: 0	Actitud: Defensiva	Posición: C4	Vive: Si
- Nombre:	Soldado5X2	Vida: 2	Ataque: 3	Defensa: 3	Velocidad: 0	Actitud: Defensiva	Posición: G2	Vive: Si
- Nombre:	Soldado5X3	Vida: 1	Ataque: 3	Defensa: 4	Velocidad: 0	Actitud: Defensiva	Posición: H5	Vive: Si
- Nombre:	Soldado5X4	Vida: 1	Ataque: 1	Defensa: 1	Velocidad: 0	Actitud: Defensiva	Posición: I4	Vive: Si
- Nombre:	Soldado5X5	Vida: 4	Ataque: 4	Defensa: 3	Velocidad: 0	Actitud: Defensiva	Posición: C7	Vive: Si
- Nombre:	Soldado5X6	Vida: 4	Ataque: 2	Defensa: 4	Velocidad: 0	Actitud: Defensiva	Posición: A2	Vive: Si
- Nombre:	Soldado5X7	Vida: 3	Ataque: 5	Defensa: 5	Velocidad: 0	Actitud: Defensiva	Posición: C6	Vive: Si

Ejercito 6 ---> 8 soldados, 24 puntos de vida, ubicado en la posición G8

Soldados de este ejército:

- Nombre:	Soldado6X1	Vida: 5	Ataque: 3	Defensa: 1	Velocidad: 0	Actitud: Defensiva	Posición: G9	Vive: Si
- Nombre:	Soldado6X2	Vida: 1	Ataque: 3	Defensa: 5	Velocidad: 0	Actitud: Defensiva	Posición: C7	Vive: Si
- Nombre:	Soldado6X3	Vida: 3	Ataque: 1	Defensa: 4	Velocidad: 0	Actitud: Defensiva	Posición: I9	Vive: Si
- Nombre:	Soldado6X4	Vida: 4	Ataque: 2	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: J7	Vive: Si
- Nombre:	Soldado6X5	Vida: 1	Ataque: 3	Defensa: 1	Velocidad: 0	Actitud: Defensiva	Posición: B5	Vive: Si
- Nombre:	Soldado6X6	Vida: 5	Ataque: 5	Defensa: 4	Velocidad: 0	Actitud: Defensiva	Posición: B7	Vive: Si
- Nombre:	Soldado6X7	Vida: 2	Ataque: 2	Defensa: 1	Velocidad: 0	Actitud: Defensiva	Posición: E4	Vive: Si
- Nombre:	Soldado6X8	Vida: 3	Ataque: 2	Defensa: 4	Velocidad: 0	Actitud: Defensiva	Posición: D3	Vive: Si

Ejercito 7 ---> 6 soldados, 16 puntos de vida, ubicado en la posición H4

Soldados de este ejército:

- Nombre:	Soldado7X1	Vida: 3	Ataque: 1	Defensa: 5	Velocidad: 0	Actitud: Defensiva	Posición: H9	Vive: Si
- Nombre:	Soldado7X2	Vida: 3	Ataque: 4	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: A7	Vive: Si
- Nombre:	Soldado7X3	Vida: 2	Ataque: 1	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: D6	Vive: Si
- Nombre:	Soldado7X4	Vida: 3	Ataque: 4	Defensa: 3	Velocidad: 0	Actitud: Defensiva	Posición: C4	Vive: Si
- Nombre:	Soldado7X5	Vida: 4	Ataque: 1	Defensa: 1	Velocidad: 0	Actitud: Defensiva	Posición: G4	Vive: Si
- Nombre:	Soldado7X6	Vida: 1	Ataque: 5	Defensa: 3	Velocidad: 0	Actitud: Defensiva	Posición: G10	Vive: Si

Ejercito 8 ---> 6 soldados, 18 puntos de vida, ubicado en la posición B7

Soldados de este ejército:

- Nombre:	Soldado8X1	Vida: 3	Ataque: 3	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: C3	Vive: Si
- Nombre:	Soldado8X2	Vida: 5	Ataque: 1	Defensa: 4	Velocidad: 0	Actitud: Defensiva	Posición: F7	Vive: Si
- Nombre:	Soldado8X3	Vida: 2	Ataque: 4	Defensa: 5	Velocidad: 0	Actitud: Defensiva	Posición: J9	Vive: Si
- Nombre:	Soldado8X4	Vida: 2	Ataque: 2	Defensa: 5	Velocidad: 0	Actitud: Defensiva	Posición: C10	Vive: Si
- Nombre:	Soldado8X5	Vida: 3	Ataque: 1	Defensa: 3	Velocidad: 0	Actitud: Defensiva	Posición: G10	Vive: Si
- Nombre:	Soldado8X6	Vida: 3	Ataque: 4	Defensa: 5	Velocidad: 0	Actitud: Defensiva	Posición: A2	Vive: Si

REINO DE FRANCIA

+ Total de ejércitos: 5

+ Total de soldados: 23

+ Total de vida: 71

Lista de ejércitos creados para este reino:

Ejercito 1 ---> 4 soldados, 17 puntos de vida, ubicado en la posición C5

Soldados de este ejército:

- Nombre: Soldado1X1	Vida: 2	Ataque: 1	Defensa: 4	Velocidad: 0	Actitud: Defensiva	Posición: B1	Vive: Si
- Nombre: Soldado1X2	Vida: 5	Ataque: 3	Defensa: 1	Velocidad: 0	Actitud: Defensiva	Posición: B10	Vive: Si
- Nombre: Soldado1X3	Vida: 5	Ataque: 5	Defensa: 3	Velocidad: 0	Actitud: Defensiva	Posición: H7	Vive: Si
- Nombre: Soldado1X4	Vida: 5	Ataque: 2	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: I8	Vive: Si

Ejercito 2 ---> 5 soldados, 17 puntos de vida, ubicado en la posición I7

Soldados de este ejército:

- Nombre: Soldado2X1	Vida: 5	Ataque: 3	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: B9	Vive: Si
- Nombre: Soldado2X2	Vida: 2	Ataque: 4	Defensa: 5	Velocidad: 0	Actitud: Defensiva	Posición: I1	Vive: Si
- Nombre: Soldado2X3	Vida: 5	Ataque: 3	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: A6	Vive: Si
- Nombre: Soldado2X4	Vida: 4	Ataque: 2	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: J2	Vive: Si
- Nombre: Soldado2X5	Vida: 1	Ataque: 2	Defensa: 4	Velocidad: 0	Actitud: Defensiva	Posición: A8	Vive: Si

Ejercito 3 ---> 8 soldados, 22 puntos de vida, ubicado en la posición C3

Soldados de este ejército:

- Nombre: Soldado3X1	Vida: 5	Ataque: 2	Defensa: 1	Velocidad: 0	Actitud: Defensiva	Posición: B4	Vive: Si
- Nombre: Soldado3X2	Vida: 1	Ataque: 4	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: E2	Vive: Si
- Nombre: Soldado3X3	Vida: 3	Ataque: 5	Defensa: 3	Velocidad: 0	Actitud: Defensiva	Posición: B1	Vive: Si
- Nombre: Soldado3X4	Vida: 4	Ataque: 4	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: J4	Vive: Si
- Nombre: Soldado3X5	Vida: 2	Ataque: 1	Defensa: 4	Velocidad: 0	Actitud: Defensiva	Posición: C8	Vive: Si
- Nombre: Soldado3X6	Vida: 1	Ataque: 3	Defensa: 1	Velocidad: 0	Actitud: Defensiva	Posición: A6	Vive: Si
- Nombre: Soldado3X7	Vida: 2	Ataque: 4	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: F5	Vive: Si
- Nombre: Soldado3X8	Vida: 4	Ataque: 3	Defensa: 3	Velocidad: 0	Actitud: Defensiva	Posición: F2	Vive: Si

Ejercito 4 ---> 2 soldados, 6 puntos de vida, ubicado en la posición H7

Soldados de este ejército:

- Nombre: Soldado4X1	Vida: 5	Ataque: 2	Defensa: 2	Velocidad: 0	Actitud: Defensiva	Posición: G4	Vive: Si
- Nombre: Soldado4X2	Vida: 1	Ataque: 5	Defensa: 4	Velocidad: 0	Actitud: Defensiva	Posición: I7	Vive: Si

Ejercito 5 ---> 4 soldados, 9 puntos de vida, ubicado en la posición E6

Soldados de este ejército:

- Nombre: Soldado5X1	Vida: 4	Ataque: 4	Defensa: 1	Velocidad: 0	Actitud: Defensiva	Posición: A7	Vive: Si
- Nombre: Soldado5X2	Vida: 1	Ataque: 4	Defensa: 1	Velocidad: 0	Actitud: Defensiva	Posición: G4	Vive: Si
- Nombre: Soldado5X3	Vida: 2	Ataque: 2	Defensa: 4	Velocidad: 0	Actitud: Defensiva	Posición: D10	Vive: Si
- Nombre: Soldado5X4	Vida: 2	Ataque: 4	Defensa: 3	Velocidad: 0	Actitud: Defensiva	Posición: A9	Vive: Si

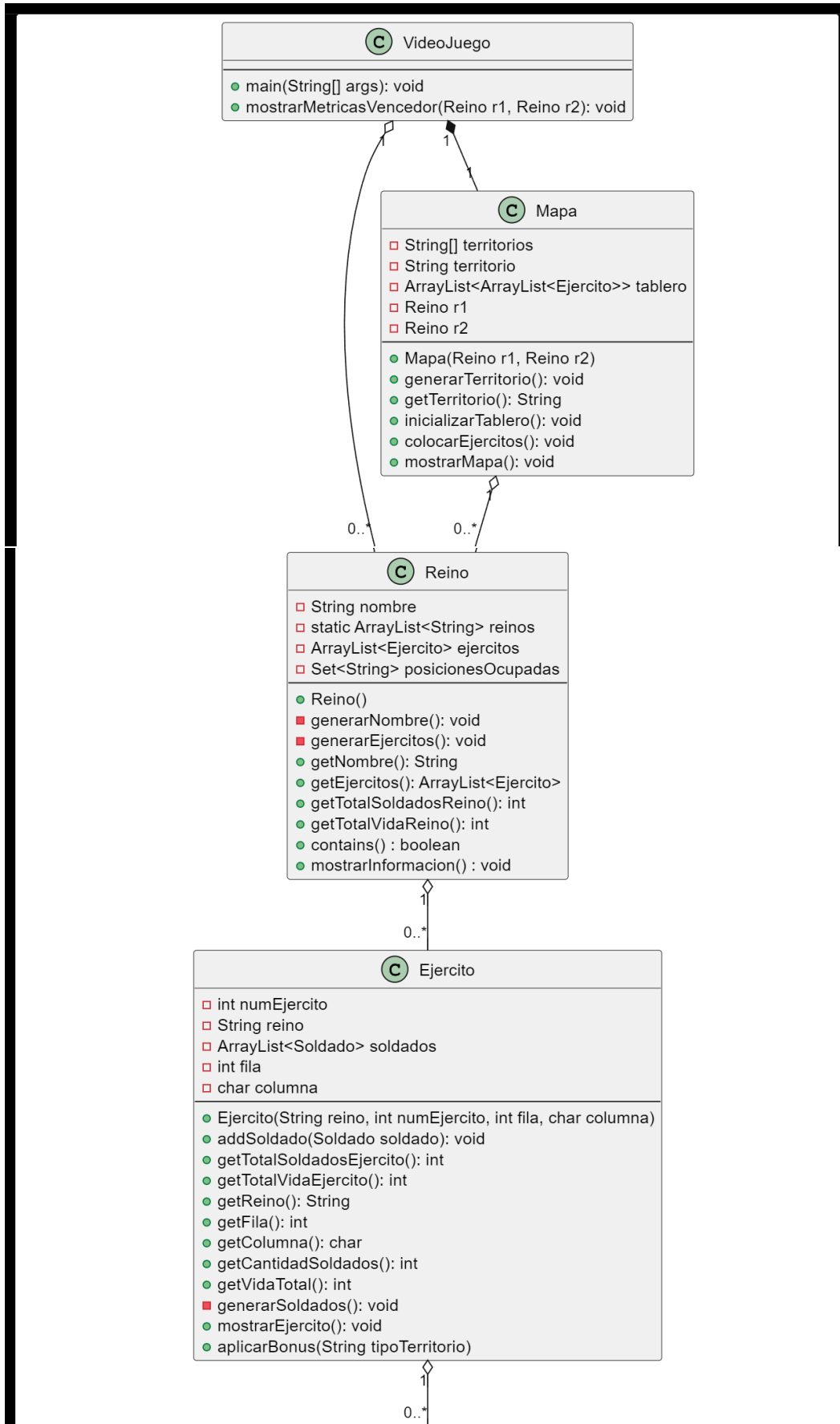
Las 3 métricas usadas para determinar al vencedor:

- Cantidad de soldados vivos
Cantidad de soldados vivos del reino de Inglaterra: 60
Cantidad de soldados vivos del reino de Francia: 23
.....
El reino de Inglaterra ha ganado la batalla.
 - Total de niveles de vida
Total de niveles de vida del reino de Inglaterra: 174
Total de niveles de vida del reino de Francia: 71
.....
El reino de Inglaterra ha ganado la batalla.
 - Probabilidades respecto a sus puntos de vida
Probabilidad de vida total del reino de Inglaterra: 71.0204081632653%
Probabilidad de vida roral del reino de Francia: 28.97959183673469%

Calculando el resultado de la batalla...

El reino de Inglaterra ha ganado la batalla.
-

Diagrama de clases UML (Con la extensión PlantUML de VS Code):



C Soldado

- ❑ String nombre
- ❑ int nivelVida
- ❑ int fila
- ❑ char columna
- ❑ int nivelAtaque
- ❑ int nivelDefensa
- ❑ int velocidad
- ❑ String actitud
- ❑ boolean vive
- ❑ int ejercito

- Soldado(String nombre, int nivelVida, int fila, char columna, int nivelAtaque, int nivelDefensa, int ejercito)
- Soldado(String nombre, int nivelVida, int fila, char columna, int ejercito)
- Soldado(String nombre, int nivelVida, int ejercito)
- getNombre(): String
- getFila(): int
- getColumna(): char
- getNivelVida(): int
- getNivelAtaque(): int
- getNivelDefensa(): int
- getVelocidad(): int
- getActitud(): String
- isVivo(): boolean
- getEjercito(): int
- setFila(int fila): void
- setColumna(char columna): void
- setNivelVida(int nivelVida): void
- setNivelAtaque(int nivelAtaque): void
- setNivelDefensa(int nivelDefensa): void
- setEjercito(int ejercito): void
- atacar(): void
- defender(): void
- huir(): void
- avanzar(): void
- retroceder(): void
- serAtacado(int daño): void
- recibirAtaque(int daño): void
- morir(): void