Universidad Nacional de San Agustín
Escuela Profesional de Ingeniería de Sistemas
Fundamentos de Programación II
Práctica de Laboratorio N° 17:
Definición de Clases de Usuario
Herencia

Nombre: Jhonatan Benjamin Mamani Céspedes **CUI: 20232188**

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

1. En la historia, los ejércitos estaban conformados por diferentes tipos de soldados, que tenían similitudes, pero también particularidades.

2. Basándose en la clase Soldado crear las clases Espadachín, Arquero y Caballero. Las tres clases heredan de la superclase Soldado, pero aumentan atributos y métodos propios, o sobrescriben métodos heredados.

3. Los espadachines tienen como atributo particular "longitud de espada" y como acción "crear un muro de escudos" que es un tipo de defensa en particular.

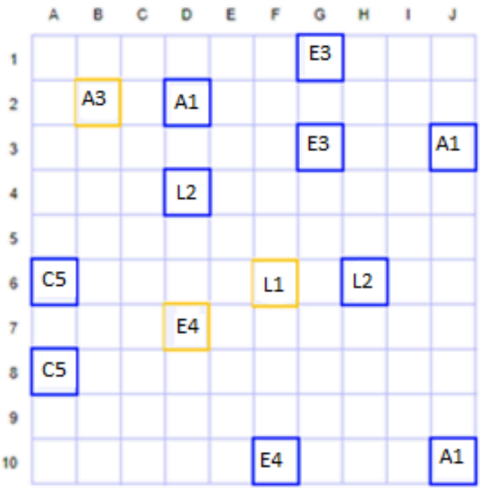
4. Los caballeros pueden alternar sus armas entre espada y lanza, además de desmontar (sólo se realiza cuando está montando e implica defender y cambiar de arma a espada), montar (sólo se realiza cuando está desmontado e implica montar, cambiar de arma a lanza y envestir). El caballero también puede envestir, ya sea montando o desmontando, cuando es desmontado equivale a atacar 2 veces, pero cuando está montando implica a atacar 3 veces.

5. Los arqueros tienen un número de flechas disponibles las cuales pueden dispararse y se agotan cuando lo hacen.

6. Basarse en los laboratorios anteriores.

7. Realizar el diagrama de clases de UML

8. Tendrá 2 Ejércitos que pueden ser constituidos sólo por espadachines, caballeros y arqueros (usar una estructura de datos por cada tipo de soldado). Cada ejército tendrá n soldados aleatorios entre 1 y 10. Cada soldado tendrá un nombre autogenerado: Espadachin0X1, Arquero1X1, Caballero2X2, etc., un valor de puntos de vida autogenerado aleatoriamente: Espadachín [3..4], Arquero [1..3] y Caballero [3..5], la fila y columna también autogenerados aleatoriamente (no puede haber 2 soldados en el mismo cuadrado) y valores autogenerados para el resto de atributos. Mostrar el tablero, distinguiendo los ejércitos y los tipos de soldados creados. Además, se debe mostrar todos los datos de todos los soldados creados para ambos ejércitos. También se mostrarán los datos del soldado con mayor vida de cada ejército, el promedio de puntos de vida de todos los soldados creados por ejército, los datos de todos los soldados por ejército en el orden que fueron creados y un ranking de poder de todos los soldados creados por ejército (del que tiene más nivel de vida al que tiene menos) usando algún algoritmo de ordenamiento. Finalmente, que muestre qué ejército ganará la batalla (indicar la métrica usada para decidir al ganador de la batalla). Hacerlo programa iterativo.



Clase Soldado.java:

```java
public class Soldado {
    public static final int MAX_SOLDADOS_POR_EJERCITO = 10;
    public static int totalSoldadosCreados = 0;
    public static int soldadosEjercito1 = 0;
    public static int soldadosEjercito2 = 0;
    private String nombre;
    private int puntosVida;
    private int fila;
    private char columna;
    private int nivelAtaque;
    private int nivelDefensa;
    private int velocidad;
    private String actitud;
    private boolean vive;
    private int ejercito;

    // Constructores sobrecargados del Soldado
    public Soldado(String nombre, int puntosVida, int fila, char columna, int nivelAtaque, int nivelDefensa, int ejercito) {
        this.nombre = nombre;
        this.puntosVida = puntosVida;
        this.fila = fila;
        this.columna = columna;
        this.nivelAtaque = nivelAtaque;
        this.nivelDefensa = nivelDefensa;
        this.velocidad = 0;
        this.actitud = "Defensiva";
        this.vive = true;
        this.ejercito = ejercito;
        totalSoldadosCreados++;
    }
    public Soldado(String nombre, int puntosVida, int fila, char columna, int ejercito) {
        this.nombre = nombre;
        this.puntosVida = puntosVida;
        this.fila = fila;
        this.columna = columna;
        this.nivelAtaque = (int)(Math.random() * 5) + 1;
```

```
        this.nivelDefensa = (int)(Math.random() * 5) + 1;
        this.velocidad = 0;
        this.actitud = "Defensiva";
        this.vive = true;
        this.ejercito = ejercito;
        totalSoldadosCreados++;
    }
    public Soldado(String nombre, int nivelVida, int ejercito) {
        this.nombre = nombre;
        this.nivelVida = nivelVida;
        this.fila = 0;
        this.columna = 'A';
        this.nivelAtaque = (int)(Math.random() * 5) + 1;
        this.nivelDefensa = (int)(Math.random() * 5) + 1;
        this.velocidad = 0;
        this.actitud = "Defensiva";
        this.vive = true;
        this.ejercito = ejercito;
        totalSoldadosCreados++;
    }

    // Getters
    public String getNombre() {
        return nombre;
    }
    public int getFila() {
        return fila;
    }
    public char getColumna() {
        return columna;
    }
    public int getNivelVida() {
        return nivelVida;
    }
    public int getNivelAtaque() {
        return nivelAtaque;
    }
    public int getNivelDefensa() {
        return nivelDefensa;
    }
    public int getVelocidad() {
        return velocidad;
    }
    public String getActitud() {
        return actitud;
    }
    public boolean isVivo() {
        return vive;
    }
    public int getEjercito() {
        return ejercito;
    }

    // Setters
    public void setFila(int fila) {
        this.fila = fila;
    }
    public void setColumna(char columna) {
        this.columna = columna;
    }
    public void setNivelVida(int nivelVida) {
        this.nivelVida = nivelVida;
    }
    public void setNivelAtaque(int nivelAtaque) {
        this.nivelAtaque = nivelAtaque;
    }
    public void setNivelDefensa(int nivelDefensa) {
        this.nivelDefensa = nivelDefensa;
    }
    public void setEjercito(int ejercito) {
        this.ejercito = ejercito;
    }

    // Métodos del UML de referencia
    public void atacar() {
        velocidad += 1;
        actitud = "Ofensiva";
        System.out.println(nombre + " ha atacado, su velocidad es ahora " + velocidad);
    }
```

```java
    public void defender() {
        actitud = "Defensiva";
        System.out.println(nombre + " está en modo defensivo.");
    }
    public void huir() {
        velocidad += 2;
        actitud = "Fuga";
        System.out.println(nombre + " está huyendo, su velocidad es ahora " + velocidad);
    }
    public void avanzar() {
        velocidad += 1;
        System.out.println(nombre + " avanza, su velocidad es ahora " + velocidad);
    }
    public void retroceder() {
        if (velocidad > 0) {
            velocidad = 0;
            actitud = "Defensiva";
            System.out.println(nombre + " se ha detenido, velocidad actual: " + velocidad);
        } else {
            velocidad -= 1;
            System.out.println(nombre + " ha retrocedido, velocidad negativa: " + velocidad);
        }
    }
    public void serAtacado(int daño) {
        recibirAtaque(daño);
    }
    public void recibirAtaque(int daño) {
        nivelVida -= daño;
        if (nivelVida <= 0) {
            nivelVida = 0;
            System.out.println(nombre + " ha muerto.");
        } else {
            System.out.println(nombre + " ha recibido " + daño + " de daño. Vida restante: " + nivelVida);
        }
    }
    public void morir() {
        vive = false;
        System.out.println(nombre + " ha muerto.");
    }

    @Override
    public String toString() {
        return "Nombre: " + nombre
            + " | Vida: " + nivelVida
            + " | Ataque: " + nivelAtaque
            + " | Defensa: " + nivelDefensa
            + " | Defensa: " + nivelDefensa
            + " | Velocidad: " + velocidad
            + " | Actitud: " + actitud
            + " | Posición: " + columna + fila
            + " | Vive: " + (vive ? "Si" : "No");
    }
}
```

Clase Espadachin.java:

```java
public class Espadachin extends Soldado {
    private double longitudEspada;

    public Espadachin(String nombre, int nivelVida, int fila, char columna, int nivelAtaque, int nivelDefensa, int ejercito, double longitudEspada) {
        super(nombre, nivelVida, fila, columna, nivelAtaque, nivelDefensa, ejercito);
        this.longitudEspada = longitudEspada;
    }

    public void crearMuroDeEscudos() {
        System.out.println(getNombre() + " ha creado un muro de escudos.");
    }

    @Override
    public String toString() {
        return super.toString() + " | Longitud Espada: " + longitudEspada;
    }
}
```

Clase Caballero.java:

```java
public class Caballero extends Soldado {
    private boolean montado;
    private String arma;

    public Caballero(String nombre, int nivelVida, int fila, char columna, int nivelAtaque, int nivelDefensa, int ejercito) {
        super(nombre, nivelVida, fila, columna, nivelAtaque, nivelDefensa, ejercito);
        this.montado = true;
        this.arma = "lanza";
    }

    public void montar() {
        if (!montado) {
            montado = true;
            arma = "lanza";
            System.out.println(getNombre() + " ha montado y cambiado a " + arma + ".");
        } else {
            System.out.println(getNombre() + " ya está montado.");
        }
    }

    public void desmontar() {
        if (montado) {
            montado = false;
            arma = "espada";
            System.out.println(getNombre() + " ha desmontado y cambiado a " + arma + ".");
        } else {
            System.out.println(getNombre() + " ya está desmontado.");
        }
    }

    public void envestir() {
        int ataques = montado ? 3 : 2;
        System.out.println(getNombre() + " realizó una envestida con " + ataques + " ataques.");
    }

    @Override
    public String toString() {
        return super.toString() + " | Montado: " + montado + " | Arma: " + arma;
    }
}
```

Clase Arquero.java:

```java
public class Arquero extends Soldado {
    private int flechasDisponibles;

    public Arquero(String nombre, int nivelVida, int fila, char columna, int nivelAtaque, int nivelDefensa, int ejercito, int flechasDisponibles) {
        super(nombre, nivelVida, fila, columna, nivelAtaque, nivelDefensa, ejercito);
        this.flechasDisponibles = flechasDisponibles;
    }

    public int getFlechasDisponibles() {
        return flechasDisponibles;
    }

    public void dispararFlecha() {
        if (flechasDisponibles > 0) {
            flechasDisponibles--;
            System.out.println(getNombre() + " ha disparado una flecha. Flechas restantes: " + flechasDisponibles);
        } else {
            System.out.println(getNombre() + " no tiene flechas disponibles para disparar.");
        }
    }

    @Override
    public void atacar() {
        if (flechasDisponibles > 0) {
            dispararFlecha();
        } else {
            System.out.println(getNombre() + " no puede atacar porque no tiene flechas.");
        }
    }
}
```

Clase Ejército.java:

```java
import java.util.*;

public class Ejercito {
    private int numEjercito;
    private ArrayList<Soldado> soldados = new ArrayList<>();
    private Set<String> posicionesOcupadas = new HashSet<>();

    public Ejercito(int numEjercito) {
        this.numEjercito = numEjercito;
        generarSoldados();
    }

    public void addSoldado(Soldado soldado) {
        soldados.add(soldado);
    }

    public int getTotalSoldadosEjercito() {
        return soldados.size();
    }

    public int getTotalVidaEjercito() {
        int total = 0;
        for (Soldado s : soldados) {
            total += s.getNivelVida();
        }
        return total;
    }

    public ArrayList<Soldado> getSoldados() {
        return soldados;
    }

    public Soldado getSoldadoConMayorVida() {
        return soldados.stream().max(Comparator.comparingInt(Soldado::getNivelVida)).orElse(null);
    }

    public double getPromedioVida() {
        return soldados.stream().mapToInt(Soldado::getNivelVida).average().orElse(0);
    }

    private void generarSoldados() {
        int n = (int) (Math.random() * Soldado.MAX_SOLDADOS_POR_EJERCITO) + 1;
        for (int i = 0; i < n; i++) {
            Soldado soldado = crearSoldadoAleatorio(i + 1);
            soldados.add(soldado);
        }
        }
    }

    private Soldado crearSoldadoAleatorio(int indice) {
        String tipo = generarTipoSoldado();
        String nombre = tipo + numEjercito + "X" + indice;

        // Generar posición única
        int fila, columna;
        do {
            fila = (int) (Math.random() * 10);
            columna = (int) (Math.random() * 10);
        } while (posicionesOcupadas.contains(fila + "," + columna));
        posicionesOcupadas.add(fila + "," + columna);

        // Crear soldado según el tipo
        switch (tipo) {
            case "Espadachin":
                return new Espadachin(nombre, generarVida(3, 4), fila + 1, (char) ('A' + columna), generarNivel(), generarNivel(), numEjercito, 1.5);
            case "Arquero":
                return new Arquero(nombre, generarVida(1, 3), fila + 1, (char) ('A' + columna), generarNivel(), generarNivel(), numEjercito, 10);
            case "Caballero":
                return new Caballero(nombre, generarVida(3, 5), fila + 1, (char) ('A' + columna), generarNivel(), generarNivel(), numEjercito);
            default:
                throw new IllegalArgumentException("Tipo de soldado desconocido: " + tipo);
        }
    }

    private String generarTipoSoldado() {
        String[] tipos = {"Espadachin", "Arquero", "Caballero"};
        return tipos[(int) (Math.random() * tipos.length)];
    }

    private int generarVida(int min, int max) {
        return (int) (Math.random() * (max - min + 1)) + min;
    }

    private int generarNivel() {
        return (int) (Math.random() * 5) + 1;
    }

    public void mostrarEjercito() {
        System.out.println("\nEjército #" + numEjercito);
        for (Soldado soldado : soldados) {
            System.out.println(" - " + soldado);
        }
    }
```

```java
    public void mostrarRanking() {
        System.out.println("\nRanking del Ejército #" + numEjercito + " (Por nivel de vida):");
        soldados.stream()
                .sorted((s1, s2) -> Integer.compare(s2.getNivelVida(), s1.getNivelVida()))
                .forEach(System.out::println);
    }
}
```

Clase Mapa.java:

```java
import java.util.*;

public class Mapa {
    private ArrayList<ArrayList<String>> tablero = new ArrayList<>();
    private Ejercito ejercito1;
    private Ejercito ejercito2;

    public Mapa(Ejercito ejercito1, Ejercito ejercito2) {
        this.ejercito1 = ejercito1;
        this.ejercito2 = ejercito2;
        inicializarTablero();
        colocarSoldados();
    }

    private void inicializarTablero() {
        for (int i = 0; i < 10; i++) {
            ArrayList<String> fila = new ArrayList<>();
            for (int j = 0; j < 10; j++) {
                fila.add("    ");
            }
            tablero.add(fila);
        }
    }

    private void colocarSoldados() {
        colocarSoldadosPorEjercito(ejercito1, 1);
        colocarSoldadosPorEjercito(ejercito2, 2);
    }

    private void colocarSoldadosPorEjercito(Ejercito ejercito, int numEjercito) {
        for (Soldado soldado : ejercito.getSoldados()) {
            int fila = soldado.getFila() - 1;
            int columna = soldado.getColumna() - 'A';

            String tipoSoldado = obtenerTipoSoldado(soldado);
            String simbolo = (numEjercito == 1 ? "[" : "<") + tipoSoldado + soldado.getNivelVida() + (numEjercito == 1 ? "]" : ">");

            tablero.get(fila).set(columna, simbolo);
        }
    }

    private String obtenerTipoSoldado(Soldado soldado) {
        if (soldado instanceof Arquero) {
            return "A";
        } else if (soldado instanceof Espadachin) {
            return "E";
        } else if (soldado instanceof Caballero) {
            return "C";
        } else {
            return "S";
        }
    }

    public void mostrarMapa() {
        System.out.println("\n═══════════════════════════════════════════════════════════════════════════════");
        System.out.println("████    ████    ████    CAMPO DE BATALLA    ████    ████    ████");
        System.out.println("═══════════════════════════════════════════════════════════════════════════════");
        System.out.println("\n* Los soldados del Ejército 1 están representados por '[tipo/Vida]'"
                           + "\n* Los del Ejército 2 por '<tipo/Vida>'\n");
        System.out.println("═══════════════════════════════════════════════════════════════════════════════");

        System.out.println("\n          A     B     C     D     E     F     G     H     I     J");
        System.out.println();
        System.out.println("       ------------------------------------------------------------------");

```

```
        for (int i = 0; i < tablero.size(); i++) {
            System.out.print("  " + (i + 1) + "\t| ");
            for (String cell : tablero.get(i)) {
                if (cell == "") {
                    cell = "     ";
                }
                System.out.print("" + cell + " | ");
            }
            System.out.println();
            System.out.println("        ------------------------------------------------------------------------");
        }
        System.out.println("\n═══════════════════════════════════════════════════════════════════════════════════════\n");
    }
}

```

Clase VideoJuego.java

```java
// LABORATORIO N° 17
// AUTOR: JHONATAN BENJAMIN MAMANI CÉSPEDES
// TIEMPO: 59 MINUTOS
public class VideoJuego {
    public static void main(String[] args) {
        Ejercito ejercito1 = new Ejercito(1);
        Ejercito ejercito2 = new Ejercito(2);

        Mapa mapa = new Mapa(ejercito1, ejercito2);

        mapa.mostrarMapa();

        System.out.println("\n--- Información del Ejército 1 ---");
        ejercito1.mostrarEjercito();
        System.out.println("\n--- Información del Ejército 2 ---");
        ejercito2.mostrarEjercito();

        System.out.println("\n--- Resultado de la Batalla ---");
        int vidaE1 = ejercito1.getTotalVidaEjercito();
        int vidaE2 = ejercito2.getTotalVidaEjercito();

        System.out.println("Vida total Ejército 1: " + vidaE1);
        System.out.println("Vida total Ejército 2: " + vidaE2);

        if (vidaE1 > vidaE2) {
            System.out.println("¡El Ejército 1 gana la batalla!");
        } else if (vidaE2 > vidaE1) {
            System.out.println("¡El Ejército 2 gana la batalla!");
        } else {
            System.out.println("¡La batalla termina en empate!");
        }
    }
}
```

Consola:

```
═══════════════════════════════════════════════════════════════
▇▇▇    ▇▇▇    ▇▇▇    C A M P O   D E   B A T A L L A    ▇▇▇    ▇▇▇    ▇▇▇
═══════════════════════════════════════════════════════════════

* Los soldados del Ejército 1 están representados por '[tipo/Vida]'
* Los del Ejército 2 por '<tipo/Vida>'

═══════════════════════════════════════════════════════════════

          A      B      C      D      E      F      G      H      I      J

        ----------------------------------------------------------------------
  1     | [E4] |      |      |      |      |      |      |      |      |      |
        ----------------------------------------------------------------------
  2     |      |      |      |      | <A3> | [A3] |      |      |      |      |
        ----------------------------------------------------------------------
  3     |      |      |      |      |      |      |      |      |      |      |
        ----------------------------------------------------------------------
  4     | [C5] |      |      |      |      |      |      |      |      |      |
        ----------------------------------------------------------------------
  5     | [C5] |      |      |      |      |      |      |      |      |      |
        ----------------------------------------------------------------------
  6     |      |      |      |      |      |      |      |      | <C5> |      |
        ----------------------------------------------------------------------
  7     |      | [E3] |      |      |      |      |      |      | [E4] |      |
        ----------------------------------------------------------------------
  8     |      | [A3] |      |      |      |      |      |      |      |      |
        ----------------------------------------------------------------------
  9     | <A2> |      |      |      |      | <A1> |      |      |      |      |
        ----------------------------------------------------------------------
  10    |      |      |      |      |      |      |      |      |      | <A2> |
        ----------------------------------------------------------------------

═══════════════════════════════════════════════════════════════


--- Información del Ejército 1 ---

Ejército #1
 - Nombre: Espadachin1X1 | Vida: 4 | Ataque: 3 | Defensa: 2 | Velocidad: 0 | Actitud: Defensiva | Posición: I7 | Vive: Si | Longitud Espada: 1.5
 - Nombre: Caballero1X2 | Vida: 5 | Ataque: 2 | Defensa: 1 | Velocidad: 0 | Actitud: Defensiva | Posición: A4 | Vive: Si | Montado: true | Arma: lanza
 - Nombre: Arquero1X3 | Vida: 3 | Ataque: 3 | Defensa: 4 | Velocidad: 0 | Actitud: Defensiva | Posición: F2 | Vive: Si
 - Nombre: Arquero1X4 | Vida: 3 | Ataque: 2 | Defensa: 1 | Velocidad: 0 | Actitud: Defensiva | Posición: B8 | Vive: Si
 - Nombre: Caballero1X5 | Vida: 5 | Ataque: 1 | Defensa: 2 | Velocidad: 0 | Actitud: Defensiva | Posición: A5 | Vive: Si | Montado: true | Arma: lanza
 - Nombre: Espadachin1X6 | Vida: 3 | Ataque: 2 | Defensa: 5 | Velocidad: 0 | Actitud: Defensiva | Posición: B7 | Vive: Si | Longitud Espada: 1.5
 - Nombre: Espadachin1X7 | Vida: 4 | Ataque: 5 | Defensa: 5 | Velocidad: 0 | Actitud: Defensiva | Posición: A1 | Vive: Si | Longitud Espada: 1.5

--- Información del Ejército 2 ---

Ejército #2
 - Nombre: Arquero2X1 | Vida: 2 | Ataque: 2 | Defensa: 2 | Velocidad: 0 | Actitud: Defensiva | Posición: J10 | Vive: Si
 - Nombre: Arquero2X2 | Vida: 1 | Ataque: 4 | Defensa: 3 | Velocidad: 0 | Actitud: Defensiva | Posición: F9 | Vive: Si
 - Nombre: Caballero2X3 | Vida: 5 | Ataque: 3 | Defensa: 4 | Velocidad: 0 | Actitud: Defensiva | Posición: I6 | Vive: Si | Montado: true | Arma: lanza
 - Nombre: Arquero2X4 | Vida: 2 | Ataque: 5 | Defensa: 1 | Velocidad: 0 | Actitud: Defensiva | Posición: A9 | Vive: Si
 - Nombre: Arquero2X5 | Vida: 3 | Ataque: 3 | Defensa: 3 | Velocidad: 0 | Actitud: Defensiva | Posición: E2 | Vive: Si

--- Resultado de la Batalla ---
Vida total Ejército 1: 27
Vida total Ejército 2: 13
¡El Ejército 1 gana la batalla!
```

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