INFORME DE LABORATORIO

(formato estudiante)

INFORMACIÓN BÁSICA					
ASIGNATURA:	*Fundamentos de la programación 2*				
TÍTULO DE LA PRÁCTICA:	*Definición de Clases de Usuario* *Clase Soldado-Miembros de clase*				
NÚMERO DE PRÁCTICA:	*13*	**AÑO LECTIVO:**	*2023*	**NRO. SEMESTRE:**	*2do Semestre*
FECHA DE PRESENTACIÓN	*27/12/2023*	**HORA DE PRESENTACIÓN**	*15/40/00*		
INTEGRANTE (s) *Juan Diego Gutiérrez Ccama*				**NOTA (0-20)**	*Nota colocada por el docente*
DOCENTE(s): *Linno Jose Pinto Oppe*					

RESULTADOS Y PRUEBAS
I. EJERCICIOS RESUELTOS:

```java
public class Soldado {
    private static int totalSoldadosCreados = 0;
    private static int sEjer1 = 0;
    private static int sEjer2 = 0;
    public static final int Soldados = 10;
    private String nombre;
    private int nivelAtaque;
    private int nivelDefensa;
    private int vidaActual;
    private int velocidad;
    private String actitud;
    private boolean vive;
    private int fila;
    private int columna;
    private int nivelVida;

    public static void sumarEjer1() {
        sEjer1++;
        totalSoldadosCreados++;
    }
    public static void sumarEjer2() {
        sEjer2++;
        totalSoldadosCreados++;
    }
    public static void restarEjer1() {
        sEjer1--;
        totalSoldadosCreados--;
    }
    public static void restarEjer2() {
        sEjer2--;
        totalSoldadosCreados--;
    }

    public void atacar() {
        avanzar();
        actitud = "ofensiva";
    }
    public void defender() {
        velocidad = 0;
        actitud = "defensiva";
    }
    public void avanzar() {
        velocidad++;
    }
    }
    public void retroceder() {
        if (velocidad > 0) {
            velocidad = 0;
            actitud = "defensiva";
        } else {
            velocidad--;
        }
    }
    public void serAtacado(int puntosDaño) {
        vidaActual -= puntosDaño;
        if (vidaActual <= 0) {
            morir();
        }
    }
    public void huir() {
        velocidad += 2;
        actitud = "fuga";
    }
    public void morir() {
        vive = false;
    }
    public static int getSoldCread() {
        return totalSoldadosCreados;
    }

    public static int getSEjer1() {
        return sEjer1;
    }

    public static int getSEjer2() {
        return sEjer2;
    }
  //zona de sets
    public void setNombre(String n){
        nombre = n;
    }
    public void setNivelAtaque(int ataque){
        nivelAtaque = ataque;
    }
    public void setNivelDefensa(int defensa){
        nivelDefensa = defensa;
    }
```

```java
    public void setVidaActual(int vidaAc) {
          vidaActual = vidaAc;
      }
    public void setFila(int f){
        fila = f;
    }
    public void setColumna(int c){
        columna = c;
    }
    public void setNivelVida(int vida){
        nivelVida = vida;
    }
    public void setVelocidad(int veloci){
        velocidad = veloci;
    }
    //Zona de gets
    public String getNombre(){
        return nombre;
    }
    public int getFila(){
        return fila;
    }
    public int getColumna(){
        return columna;
    }
    public int getNivelVida(){
        return nivelVida;
    }
    public int getNivelAtaque(){
        return nivelAtaque;
    }
    public int getNivelDefensa(){
        return nivelDefensa;
    }
    public int getVelocidad(){
        return velocidad;
    }
    public int getVidaActual() {
        return vidaActual;
    }
}
```

Soldado.java | VideoJuego.java

```java
import java.util.*;
import java.util.ArrayList;
public class VideoJuego {
    private static ArrayList<Soldado> ejercito1 = new ArrayList<>();
    private static ArrayList<Soldado> ejercito2 = new ArrayList<>();
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int opcion = 0;
        while (opcion != 3) {
            System.out.println("Menú:");
            System.out.println("1. Juego Rápido");
            System.out.println("2. Personalizado");
            System.out.println("3. Salir");
            System.out.print("Seleccione una opción: ");
            opcion = sc.nextInt();
            switch (opcion) {
                case 1:
                    juegoRapido();
                    break;
                case 2:
                    gestionarEjercitoPersonalizado(sc, ejercito1, ejercito2);
                    break;
                case 3:
                    System.out.println("¡Hasta luego!");
                    break;
                default:
                    System.out.println("Opción no válida. Por favor, elija una opción válida.");
                    break;
            }
        }
    }
    public static void juegoRapido() {
        Scanner sc = new Scanner(System.in);
        Soldado[][] tablero = new Soldado[10][10];
        int n = (int) (Math.random() * 10) + 1;
        int x = (int) (Math.random() * 10) + 1;
        // Ejercito1
        llenarSoldados1(ejercito1, n);
        System.out.println("\nEjercito-1");
        imprimirSoldados(ejercito1);
        System.out.println("\nEl soldado con mayor vida es: ");
        Soldado mayorVida = soldadoConMayorVida(ejercito1);
        System.out.println("Nombre: " + mayorVida.getNombre());
```

```
        System.out.println("Vida: " + mayorVida.getNivelVida());
        double promedioVida = calcularPromedioDeVida(ejercito1);
        System.out.println("\nPromedio de vida de todos los soldados: " + promedioVida);
        int sumaVidas = calcularSumaDeVidas(ejercito1);
        System.out.println("\nEl nivel de vida de todo el ejército es: " + sumaVidas);
        ordenamientoBurbuja(ejercito1);
        System.out.println("\n--------------------");
        for (Soldado soldado : ejercito1) {
            int fila = soldado.getFila() - 1;
            int columna = soldado.getColumna() - 1;
            tablero[fila][columna] = soldado;
        }
        //Ejercito2
        llenarSoldados2(ejercito2, x);
        System.out.println("\nEjercito-2");
        imprimirSoldados(ejercito2);
        System.out.println("\nEl soldado con mayor vida es: ");
        Soldado mayorV = soldadoConMayorVida(ejercito2);
        System.out.println("Nombre: " + mayorV.getNombre());
        System.out.println("Vida: " + mayorV.getNivelVida());
        double promedioV = calcularPromedioDeVida(ejercito2);
        System.out.println("\nPromedio de vida de todos los soldados: " + promedioV);
        int sumaV = calcularSumaDeVidas(ejercito2);
        System.out.println("\nEl nivel de vida de todo el ejército es: " + sumaV);
        ordenamientoSeleccion(ejercito2);
        System.out.println("\nCantidad total de objetos Soldado creados: " + Soldado.getSoldCread());
        for (Soldado soldado : ejercito2) {
            int fila = soldado.getFila() - 1;
            int columna = soldado.getColumna() - 1;
            tablero[fila][columna] = soldado;
        }
        boolean juegoActivo = true;
        while (juegoActivo) {
            System.out.println("\nTablero de juego:");
            System.out.println("\nCantidad de soldados en el Ejército 1: " + Soldado.getSEjer1());
            System.out.println("Cantidad de soldados en el Ejército 2: " + Soldado.getSEjer2());
            System.out.println("Cantidad de soldados en total: " + Soldado.getSoldCread());
            imprimirTablero(tablero, ejercito1, ejercito2);
            System.out.println("Turno del Jugador 1 - S ");
            moverSoldado(sc, tablero, ejercito1, ejercito2);
            if (ejercito2.isEmpty()) {
                System.out.println("¡El Jugador 1 ha ganado!");
                break;
            }
            System.out.println("Tablero de juego:");
            System.out.println("\nCantidad de soldados en el Ejército 1: " + Soldado.getSEjer1());
            System.out.println("Cantidad de soldados en el Ejército 2: " + Soldado.getSEjer2());
            System.out.println("Cantidad de soldados en total: " + Soldado.getSoldCread());
            imprimirTablero(tablero, ejercito1, ejercito2);
            System.out.println("Turno del Jugador 2 - C ");
            moverSoldado(sc, tablero, ejercito2, ejercito1);
            if (ejercito1.isEmpty()) {
                System.out.println("¡El Jugador 2 ha ganado!");
                break;
            }
        }
    }
    public static void imprimirTablero(Soldado[][] tablero, ArrayList<Soldado> ejercito1, ArrayList<Soldado> ejercito2) {
        System.out.print("  ");
        for (char columna = 'A'; columna <= 'J'; columna++) {
            System.out.print("  " + columna);
        }
        System.out.println();
        System.out.println("    __ __ __ __ __ __ __ __ __ __ ");
        for (int i = 0; i < tablero.length; i++) {
            System.out.print((i + 1 < 10 ? " " : "") + (i + 1) );
            for (int j = 0; j < tablero[i].length; j++) {
                if (tablero[i][j] != null) {
                    Soldado soldado = tablero[i][j];
                    char letra = (ejercito1.contains(soldado)) ? 's' : 'c';
                    System.out.print("|" + letra + soldado.getNivelVida());
                } else {
                    System.out.print("|__");
                }
            }
            System.out.println("|");
        }
    }
    public static void llenarSoldados1(ArrayList<Soldado> soldados, int n) {
        int cantidadSoldadosRestantes = Soldado.Soldados - soldados.size();
        if (n > cantidadSoldadosRestantes) {
            n = cantidadSoldadosRestantes;
            System.out.println("Se han creado " + n + " soldados para el Ejército 1.");
        }
        for (int i = 0; i < n; i++) {
            int x, d, a, f, c, v;
```

```
            String nombre;
            boolean casillaOcupada;
            while (true) {
                casillaOcupada = false;
                x = (int)(Math.random() * 5 + 1);
                d = (int)(Math.random() * 5 + 1);
                a = (int)(Math.random() * 5 + 1);
                f = (int)(Math.random() * 5 + 1);
                c = (int)(Math.random() * 5 + 1);
                v = (int)(Math.random() * 5 + 1);
                nombre = "soldado" + i + "X1";
                for (Soldado s : soldados) {
                    if (s.getFila() == f && s.getColumna() == c) {
                        casillaOcupada = true;
                        break;
                    }
                }
                if (!casillaOcupada) {
                    Soldado soldado = new Soldado();
                    soldado.setNombre(nombre);
                    soldado.setFila(f);
                    soldado.setColumna(c);
                    soldado.setNivelAtaque(a);
                    soldado.setNivelDefensa(d);
                    soldado.setNivelVida(x);
                    soldado.setVelocidad(v);
                    soldados.add(soldado);
                    Soldado.sumarEjer1();
                    break;
                }
            }
        }
    }
    public static void llenarSoldados2(ArrayList<Soldado> soldados, int n) {
        int cantidadSoldadosRestantes = Soldado.Soldados - soldados.size();
        if (n > cantidadSoldadosRestantes) {
            n = cantidadSoldadosRestantes;
            System.out.println("Se han creado " + n + " soldados para el Ejército 2.");
        }
        for (int i = 0; i < n; i++) {
            int x, d, a, f, c, v;
            String nombre;
            boolean casillaOcupada;
            while (true) {
                casillaOcupada = false;
                x = (int)(Math.random() * 5 + 1);
                d = (int)(Math.random() * 5 + 1);
                a = (int)(Math.random() * 5 + 1);
                f = (int)(Math.random() * 5 + 1);
                c = (int)(Math.random() * 5 + 1);
                v = (int)(Math.random() * 5 + 1);
                nombre = "soldado" + i + "X2";
                for (Soldado s : soldados) {
                    if (s.getFila() == f && s.getColumna() == c) {
                        casillaOcupada = true;
                        break;
                    }
                }
                if (!casillaOcupada) {
                    Soldado soldado = new Soldado();
                    soldado.setNombre(nombre);
                    soldado.setFila(f);
                    soldado.setColumna(c);
                    soldado.setNivelAtaque(a);
                    soldado.setNivelDefensa(d);
                    soldado.setNivelVida(x);
                    soldado.setVelocidad(v);
                    soldados.add(soldado);
                    Soldado.sumarEjer2();
                    break;
                }
            }
        }
    }
    public static void imprimirSoldados(ArrayList<Soldado> ejercito) {
        if (ejercito == ejercito1) {
            System.out.println("Cantidad de soldados en el Ejército 1: " + Soldado.getSEjer1());
        } else if (ejercito == ejercito2) {
            System.out.println("Cantidad de soldados en el Ejército 2: " + Soldado.getSEjer2());
        }
        for (Soldado soldado : ejercito) {
            System.out.println("nombre: " + soldado.getNombre());
            System.out.println("fila: " + soldado.getFila());
            System.out.println("columna: " + soldado.getColumna());
            System.out.println("vida: " + soldado.getNivelVida());
            System.out.println("nivel de Ataque: " + soldado.getNivelAtaque());
            System.out.println("nivel de Defensa: " + soldado.getNivelDefensa());
            System.out.println("velocidad: " + soldado.getVelocidad());
        }
        System.out.println();
    }
    public static Soldado soldadoConMayorVida(ArrayList<Soldado> ejercito) {
        Soldado mayorVida = null;
        int maxVida = 0;
        for (Soldado soldado : ejercito) {
            if (soldado.getNivelVida() > maxVida) {
                maxVida = soldado.getNivelVida();
                mayorVida = soldado;
```

```
            }
        }
        return mayorVida;
    }
    public static double calcularPromedioDeVida(ArrayList<Soldado> ejercito) {
        int totalVida = 0;
        for (Soldado soldado : ejercito) {
            totalVida += soldado.getNivelVida();
        }
        return (double) totalVida / ejercito.size();
    }
    public static int calcularSumaDeVidas(ArrayList<Soldado> ejercito) {
        int sumaVidas = 0;
        for (Soldado soldado : ejercito) {
            sumaVidas += soldado.getNivelVida();
        }
        return sumaVidas;
    }
    public static void ordenamientoBurbuja(ArrayList<Soldado> ejercito) {
        System.out.println("\nRanking de poder por ordenamiento burbuja:");
        int n = ejercito.size();
        boolean ordenado;
        for (int i = 0; i < n - 1; i++) {
            ordenado = false;
            for (int j = 0; j < n - 1 - i; j++) {
                if (ejercito.get(j).getNivelVida() < ejercito.get(j + 1).getNivelVida()) {
                    Soldado temp = ejercito.get(j);
                    ejercito.set(j, ejercito.get(j + 1));
                    ejercito.set(j + 1, temp);
                    ordenado = true;
                }
            }
            if (!ordenado) {
                break;
            }
        }
        for (int i = 0; i < ejercito.size(); i++) {
            System.out.println("Posición " + (i + 1) + ": " + ejercito.get(i).getNombre() + " - Vida: " + ejercito.get(i).getNivelVida());
        }
    }
    public static void ordenamientoSeleccion(ArrayList<Soldado> ejercito) {
        System.out.println("\nRanking de poder por ordenamiento de selección:");
        int n = ejercito.size();
        for (int i = 0; i < n - 1; i++) {
            int max = i;
            for (int j = i + 1; j < n; j++) {
                if (ejercito.get(j).getNivelVida() > ejercito.get(max).getNivelVida()) {
                    max = j;
                }
            }
            Soldado temp = ejercito.get(i);
            ejercito.set(i, ejercito.get(max));
            ejercito.set(max, temp);
        }
        for (int i = 0; i < ejercito.size(); i++) {
            System.out.println("Posición " + (i + 1) + ": " + ejercito.get(i).getNombre() + " - Vida: " + ejercito.get(i).getNivelVida());
        }
    }
    public static void moverSoldado(Scanner sc, Soldado[][] tablero, ArrayList<Soldado> ejercitoMov, ArrayList<Soldado> ejercitoOtro) {
        System.out.print("Ingrese la coordenada del soldado a mover (por ejemplo, d3): ");
        String coordenada = sc.next();
        int filaMov = Integer.parseInt(coordenada.substring(1)) - 1;
        char columnaChar = coordenada.charAt(0);
        int columnaMov = columnaChar - 'a';
        if (filaMov < 0 || filaMov >= tablero.length || columnaMov < 0 || columnaMov >= tablero[0].length) {
            System.out.println("Movimiento inválido: coordenada fuera del tablero.");
            return;
        }
        Soldado soldado = tablero[filaMov][columnaMov];
        if (soldado == null || !ejercitoMov.contains(soldado)) {
            System.out.println("Movimiento inválido: no hay un soldado del ejército correspondiente en esa posición.");
            return;
        }
        System.out.print("Ingrese la dirección del movimiento (w: arriba, a: izquierda, s: abajo, d: derecha): ");
        char direccion = sc.next().charAt(0);
        int nuevaFila = filaMov;
        int nuevaColumna = columnaMov;
        if (direccion == 'w') {
            nuevaFila--;
        } else if (direccion == 'a') {
            nuevaColumna--;
        } else if (direccion == 's') {
            nuevaFila++;
        } else if (direccion == 'd') {
            nuevaColumna++;
        } else {
            System.out.println("Movimiento inválido: dirección no válida.");
            return;
        }
        if (nuevaFila < 0 || nuevaFila >= tablero.length || nuevaColumna < 0 || nuevaColumna >= tablero[0].length) {
            System.out.println("Movimiento inválido: movimiento fuera del tablero.");
            return;
        }
        if (tablero[nuevaFila][nuevaColumna] != null) {
            Soldado soldadoEnNuevaPos = tablero[nuevaFila][nuevaColumna];
            if (ejercitoOtro.contains(soldadoEnNuevaPos)) {
```

```
            double probabilidadSoldado = soldado.getNivelVida() * 100.0 / (soldado.getNivelVida() + soldadoEnNuevaPos.getNivelVida());
            double probabilidadEnemigo = soldadoEnNuevaPos.getNivelVida() * 100.0 / (soldado.getNivelVida() + soldadoEnNuevaPos.getNivelVida());
            Soldado ganadorBatalla = batalla(soldado, soldadoEnNuevaPos);
            System.out.println("¡Batalla!");
            System.out.println(soldado.getNombre() + " - Vida: " + soldado.getNivelVida() + " - Probabilidad de victoria: " + probabilidadSoldado + "%");
            System.out.println(soldadoEnNuevaPos.getNombre() + " - Vida: " + soldadoEnNuevaPos.getNivelVida() + " - Probabilidad de victoria: " + probabilidadEnemigo + "%");

            if (ganadorBatalla == soldado) {
                tablero[nuevaFila][nuevaColumna] = soldado;
                tablero[filaMov][columnaMov] = null;
                ejercitoOtro.remove(soldadoEnNuevaPos);
                soldado.setNivelVida(soldado.getNivelVida() + 1);
                if (ejercitoOtro == ejercito1) {
                    Soldado.restarEjer1();
                } else {
                    Soldado.restarEjer2();
                }
            } else {
                tablero[filaMov][columnaMov] = null;
                ejercitoMov.remove(soldado);
                soldadoEnNuevaPos.setNivelVida(soldadoEnNuevaPos.getNivelVida() + 1);
            }
            System.out.println("Ganador: " + ganadorBatalla.getNombre() + " - Vida: " + ganadorBatalla.getNivelVida());
            if (ejercitoOtro.isEmpty()) {
                if (ejercitoOtro == ejercito1) {
                    if (Soldado.getSEjer1() == 0) {
                        System.out.println("¡El Ejército 2 es el ganador del juego!");
                    } else {
                        System.out.println("¡El Ejército 1 es el ganador del juego!");
                    }
                }
            }
            return;
        } else {
            System.out.println("Movimiento inválido: ya hay un soldado en la nueva posición.");
            return;
        }
    }
    tablero[filaMov][columnaMov] = null;
    tablero[nuevaFila][nuevaColumna] = soldado;
}
public static Soldado batalla(Soldado soldado1, Soldado soldado2) {
    int totalVida = soldado1.getNivelVida() + soldado2.getNivelVida();
    double porcentaje1 = (soldado1.getNivelVida() * 100.0) / totalVida;
    double porcentaje2 = (soldado2.getNivelVida() * 100.0) / totalVida;
    int numeroAleatorio = (int) (Math.random() * 100) + 1;
    if (numeroAleatorio <= porcentaje1) {
        return soldado1;
    } else {
        return soldado2;
    }
}
public static void iCaracteristicas(Soldado soldado) {
    System.out.println("Características del soldado:");
    System.out.println("Nombre: " + soldado.getNombre());
    System.out.println("Fila: " + soldado.getFila());
    System.out.println("Columna: " + soldado.getColumna());
    System.out.println("Vida: " + soldado.getNivelVida());
    System.out.println("Nivel de Ataque: " + soldado.getNivelAtaque());
    System.out.println("Nivel de Defensa: " + soldado.getNivelDefensa());
    System.out.println("Velocidad: " + soldado.getVelocidad());
}
public static void gestionarEjercitoPersonalizado(Scanner sc, ArrayList<Soldado> ejercito1, ArrayList<Soldado> ejercito2) {
    int ejercitoSeleccionado;
    while (true) {
        System.out.println("Seleccione el ejército a gestionar (1 o 2): ");
        ejercitoSeleccionado = sc.nextInt();
        if (ejercitoSeleccionado == 1 || ejercitoSeleccionado == 2) {
            break;
        }
    }
    ArrayList<Soldado> ejercitoActual = (ejercitoSeleccionado == 1) ? ejercito1 : ejercito2;
    int opcion = 0;
    while (opcion != 11) {
        System.out.println("Menú de gestión de ejército " + ejercitoSeleccionado + ":");
        System.out.println("1. Crear Soldado");
        System.out.println("2. Eliminar Soldado");
        System.out.println("3. Clonar Soldado");
        System.out.println("4. Modificar Soldado");
        System.out.println("5. Comparar Soldados");
        System.out.println("6. Intercambiar Soldados");
        System.out.println("7. Ver Soldado (Búsqueda por nombre)");
        System.out.println("8. Ver Ejército");
        System.out.println("9. Sumar Niveles");
        System.out.println("10. Jugar");
        System.out.println("11. Volver");
        System.out.print("Seleccione una opción: ");
        opcion = sc.nextInt();
        switch (opcion) {
            case 1:
                crearSoldado(sc, ejercitoActual);
                break;
            case 2:
                eliminarSoldado(sc, ejercitoActual);
                break;
            case 3:
                clonarSoldado(sc, ejercitoActual);
                break;
```

```
422                 case 4:
423                     modificarSoldado(sc, ejercitoActual);
424                     break;
425                 case 5:
426                     compararSoldados(sc, ejercitoActual);
427                     break;
428                 case 6:
429                     intercambiarSoldados(sc, ejercitoActual);
430                     break;
431                 case 7:
432                     buscarSoldadoPorNombre(sc, ejercitoActual);
433                     break;
434                 case 8:
435                     verEjercito(ejercitoActual);
436                     break;
437                 case 9:
438                     sumarNiveles(ejercitoActual);
439                     break;
440                 case 10:
441                     jugar();
442                     break;
443                 case 11:
444                     System.out.println("Volviendo al menú principal...");
445                     break;
446                 default:
447                     System.out.println("Opción no válida. Por favor, elija una opción válida.");
448             }
449         }
450     }
451     public static void crearSoldado(Scanner sc, ArrayList<Soldado> ejercito) {
452         if (ejercito.size() >= 10) {
453             System.out.println("El ejército está completo. No se pueden agregar más soldados.");
454             return;
455         }
456         int numeroSoldado = ejercito.size() + 1;
457         String nombre = "soldado" + numeroSoldado + (ejercito == ejercito1 ? "X1" : "X2");
458         int x = (int)(Math.random() * 5 + 1);
459         int d = (int)(Math.random() * 5 + 1);
460         int a = (int)(Math.random() * 5 + 1);
461         int f = (int)(Math.random() * 5 + 1);
462         int c = (int)(Math.random() * 5 + 1);
463         int v = (int)(Math.random() * 5 + 1);
464         Soldado soldado = new Soldado();
465         soldado.setNombre(nombre);
466         soldado.setFila(f);
467         soldado.setColumna(c);
468         soldado.setNivelAtaque(a);
469         soldado.setNivelDefensa(d);
470         soldado.setNivelVida(x);
471         soldado.setVelocidad(v);
472         ejercito.add(soldado);;
473         System.out.println("Soldado " + soldado.getNombre() + " creado y agregado al ejército.");
474     }
475     public static void eliminarSoldado(Scanner sc, ArrayList<Soldado> ejercito) {
476         if (ejercito.isEmpty()) {
477             System.out.println("El ejército está vacío. No se pueden eliminar soldados.");
478             return;
479         }
480         System.out.print("Ingrese el nombre del soldado que deseas eliminar: ");
481         String nombreSoldado = sc.next();
482         boolean encontrado = false;
483         for (Soldado soldado : ejercito) {
484             if (soldado.getNombre().equals(nombreSoldado)) {
485                 ejercito.remove(soldado);
486                 System.out.println("Soldado " + nombreSoldado + " eliminado del ejército.");
487                 encontrado = true;
488                 break;
489             }
490         }
491         if (!encontrado) {
492             System.out.println("Soldado no encontrado en el ejército.");
493         }
494     }
495     public static void clonarSoldado(Scanner sc, ArrayList<Soldado> ejercito) {
496         if (ejercito.isEmpty()) {
497             System.out.println("El ejército está vacío. No se pueden clonar soldados.");
498             return;
499         }
500         System.out.print("Ingrese el nombre del soldado que deseas clonar: ");
501         String nombreSoldado = sc.next();
502         Soldado soldadoOriginal = null;
503         for (Soldado soldado : ejercito) {
504             if (soldado.getNombre().equals(nombreSoldado)) {
505                 soldadoOriginal = soldado;
506                 break;
507             }
```

```
        }
        if (soldadoOriginal == null) {
            System.out.println("Soldado no encontrado en el ejército.");
            return;
        }
        if (ejercito.size() >= 10) {
            System.out.println("El ejército ya tiene el máximo de soldados (10). No se pueden clonar más.");
            return;
        }
        Soldado copiaSoldado = new Soldado();
        copiaSoldado.setNombre(soldadoOriginal.getNombre());
        copiaSoldado.setNivelAtaque(soldadoOriginal.getNivelAtaque());
        copiaSoldado.setNivelDefensa(soldadoOriginal.getNivelDefensa());
        copiaSoldado.setNivelVida(soldadoOriginal.getNivelVida());
        copiaSoldado.setVelocidad(soldadoOriginal.getVelocidad());
        copiaSoldado.setFila(soldadoOriginal.getFila());
        copiaSoldado.setColumna(soldadoOriginal.getColumna());
        ejercito.add(copiaSoldado);
        System.out.println("Soldado " + copiaSoldado.getNombre() + " clonado y añadido al ejército.");
    }
    public static void modificarSoldado(Scanner sc, ArrayList<Soldado> ejercito) {
        if (ejercito.isEmpty()) {
            System.out.println("El ejército está vacío. No se pueden modificar soldados.");
            return;
        }
        System.out.print("Ingrese el nombre del soldado que deseas modificar: ");
        String nombreSoldado = sc.next();
        Soldado soldadoAModificar = null;
        for (Soldado soldado : ejercito) {
            if (soldado.getNombre().equals(nombreSoldado)) {
                soldadoAModificar = soldado;
                break;
            }
        }
        if (soldadoAModificar == null) {
            System.out.println("Soldado no encontrado en el ejército.");
            return;
        }
        System.out.println("Seleccione qué atributo deseas modificar:");
        System.out.println("1. Nivel de Ataque");
        System.out.println("2. Nivel de Defensa");
        System.out.println("3. Vida Actual");
        int opcionAtributo = sc.nextInt();
        switch (opcionAtributo) {
            case 1:
                System.out.print("Ingrese el nuevo nivel de ataque: ");
                int nuevoNivelAtaque = sc.nextInt();
                soldadoAModificar.setNivelAtaque(nuevoNivelAtaque);
                System.out.println("Nivel de Ataque modificado.");
                break;
            case 2:
                System.out.print("Ingrese el nuevo nivel de defensa: ");
                int nuevoNivelDefensa = sc.nextInt();
                soldadoAModificar.setNivelDefensa(nuevoNivelDefensa);
                System.out.println("Nivel de Defensa modificado.");
                break;
            case 3:
                System.out.print("Ingrese la nueva vida actual: ");
                int nuevaVidaActual = sc.nextInt();
                soldadoAModificar.setVidaActual(nuevaVidaActual);
                System.out.println("Vida Actual modificada.");
                break;
            default:
                System.out.println("Opción no válida. No se ha modificado ningún atributo.");
        }
    }
    public static void compararSoldados(Scanner sc, ArrayList<Soldado> ejercito) {
        if (ejercito.isEmpty()) {
            System.out.println("El ejército está vacío. No se pueden comparar soldados.");
            return;
        }
        System.out.print("Ingrese el nombre del primer soldado que deseas comparar: ");
        String nombreSoldado1 = sc.next();
        System.out.print("Ingrese el nombre del segundo soldado que deseas comparar: ");
        String nombreSoldado2 = sc.next();
        Soldado soldado1 = null;
        Soldado soldado2 = null;
        for (Soldado soldado : ejercito) {
            if (soldado.getNombre().equals(nombreSoldado1)) {
                soldado1 = soldado;
            }
            if (soldado.getNombre().equals(nombreSoldado2)) {
                soldado2 = soldado;
            }
        }
        if (soldado1 == null || soldado2 == null) {
```

```
            System.out.println("Al menos uno de los soldados no se encuentra en el ejército.");
            return;
        }
        boolean sonIguales = (
            soldado1.getNivelVida() == soldado2.getNivelVida() &&
            soldado1.getNivelAtaque() == soldado2.getNivelAtaque() &&
            soldado1.getNivelDefensa() == soldado2.getNivelDefensa() &&
            soldado1.getVelocidad() == soldado2.getVelocidad()
        );
        if (sonIguales) {
            System.out.println("Los soldados son idénticos en los atributos seleccionados.");
        } else {
            System.out.println("Los soldados son diferentes en al menos uno de los atributos seleccionados.");
        }
    }
    public static void intercambiarSoldados(Scanner sc, ArrayList<Soldado> ejercito) {
        if (ejercito.isEmpty()) {
            System.out.println("El ejército está vacío. No se pueden intercambiar soldados.");
            return;
        }
        System.out.print("Ingrese el nombre del primer soldado que desea intercambiar: ");
        String nombreSoldado1 = sc.next();
        System.out.print("Ingrese el nombre del segundo soldado que desea intercambiar: ");
        String nombreSoldado2 = sc.next();
        Soldado soldado1 = null;
        Soldado soldado2 = null;
        for (Soldado soldado : ejercito) {
            if (soldado.getNombre().equals(nombreSoldado1)) {
                soldado1 = soldado;
            }
            if (soldado.getNombre().equals(nombreSoldado2)) {
                soldado2 = soldado;
            }
        }
        if (soldado1 == null || soldado2 == null) {
            System.out.println("Al menos uno de los soldados no se encuentra en el ejército.");
            return;
        }
        int indiceSoldado1 = ejercito.indexOf(soldado1);
        int indiceSoldado2 = ejercito.indexOf(soldado2);
        ejercito.set(indiceSoldado1, soldado2);
        ejercito.set(indiceSoldado2, soldado1);
        System.out.println("Los soldados han sido intercambiados con éxito.");
    }
    public static void buscarSoldadoPorNombre(Scanner sc, ArrayList<Soldado> ejercito) {
        if (ejercito.isEmpty()) {
            System.out.println("El ejército está vacío. No se pueden buscar soldados.");
            return;
        }
        System.out.print("Ingrese el nombre del soldado que desea buscar: ");
        String nombreSoldado = sc.next();
        Soldado soldadoEncontrado = null;
        for (Soldado soldado : ejercito) {
            if (soldado.getNombre().equals(nombreSoldado)) {
                soldadoEncontrado = soldado;
                break;
            }
        }
        if (soldadoEncontrado != null) {
            iCaracteristicas(soldadoEncontrado);
        } else {
            System.out.println("Soldado no encontrado en el ejército.");
        }
    }
    public static void verEjercito(ArrayList<Soldado> ejercito) {
        if (ejercito.isEmpty()) {
            System.out.println("El ejército está vacío.");
            return;
        }
        System.out.println("Soldados en el ejército:");
        for (Soldado soldado : ejercito) {
            iCaracteristicas(soldado);
            System.out.println("--------------");
        }
    }
    public static void sumarNiveles(ArrayList<Soldado> ejercito) {
        if (ejercito.isEmpty()) {
            System.out.println("El ejército está vacío. No se pueden sumar niveles.");
            return;
        }
        int sumaNivelVida = 0;
        int sumaNivelAtaque = 0;
        int sumaNivelDefensa = 0;
        int sumaVelocidad = 0;
        for (Soldado soldado : ejercito) {
            sumaNivelVida += soldado.getNivelVida();
```

```java
            sumaNivelAtaque += soldado.getNivelAtaque();
            sumaNivelDefensa += soldado.getNivelDefensa();
            sumaVelocidad += soldado.getVelocidad();
        }
        System.out.println("Suma de niveles en el ejército:");
        System.out.println("Suma de Nivel de Vida: " + sumaNivelVida);
        System.out.println("Suma de Nivel de Ataque: " + sumaNivelAtaque);
        System.out.println("Suma de Nivel de Defensa: " + sumaNivelDefensa);
        System.out.println("Suma de Velocidad: " + sumaVelocidad);
    }
    public static void jugar() {
        Scanner sc = new Scanner(System.in);
        Soldado[][] tablero = new Soldado[10][10];
        int n = (int) (Math.random() * 10) + 1;
        int x = (int) (Math.random() * 10) + 1;
        // Ejercito1
        llenarSoldados1(ejercito1, n);
        System.out.println("\nEjercito-1");
        imprimirSoldados(ejercito1);
        System.out.println("\nEl soldado con mayor vida es: ");
        Soldado mayorVida = soldadoConMayorVida(ejercito1);
        System.out.println("Nombre: " + mayorVida.getNombre());
        System.out.println("Vida: " + mayorVida.getNivelVida());
        double promedioVida = calcularPromedioDeVida(ejercito1);
        System.out.println("\nPromedio de vida de todos los soldados: " + promedioVida);
        int sumaVidas = calcularSumaDeVidas(ejercito1);
        System.out.println("\nEl nivel de vida de todo el ejército es: " + sumaVidas);
        ordenamientoBurbuja(ejercito1);
        System.out.println("\n--------------------");
        for (Soldado soldado : ejercito1) {
            int fila = soldado.getFila() - 1;
            int columna = soldado.getColumna() - 1;
            tablero[fila][columna] = soldado;
        }
        //Ejercito2
        llenarSoldados2(ejercito2, x);
        System.out.println("\nEjercito-2");
        imprimirSoldados(ejercito2);
        System.out.println("\nEl soldado con mayor vida es: ");
        Soldado mayorV = soldadoConMayorVida(ejercito2);
        System.out.println("Nombre: " + mayorV.getNombre());
        System.out.println("Vida: " + mayorV.getNivelVida());
        double promedioV = calcularPromedioDeVida(ejercito2);
        double promedioV = calcularPromedioDeVida(ejercito2);
        System.out.println("\nPromedio de vida de todos los soldados: " + promedioV);
        int sumaV = calcularSumaDeVidas(ejercito2);
        System.out.println("\nEl nivel de vida de todo el ejército es: " + sumaV);
        ordenamientoSeleccion(ejercito2);
        System.out.println("\nCantidad total de objetos Soldado creados: " + Soldado.getSoldCread());
        for (Soldado soldado : ejercito2) {
            int fila = soldado.getFila() - 1;
            int columna = soldado.getColumna() - 1;
            tablero[fila][columna] = soldado;
        }
        boolean juegoActivo = true;
        while (juegoActivo) {
            System.out.println("\nTablero de juego:");
            System.out.println("\nCantidad de soldados en el Ejército 1: " + Soldado.getSEjer1());
            System.out.println("Cantidad de soldados en el Ejército 2: " + Soldado.getSEjer2());
            System.out.println("Cantidad de soldados en total: " + Soldado.getSoldCread());
            imprimirTablero(tablero, ejercito1, ejercito2);
            System.out.println("Turno del Jugador 1 - S ");
            moverSoldado(sc, tablero, ejercito1, ejercito2);
            if (ejercito2.isEmpty()) {
                System.out.println("¡El Jugador 1 ha ganado!");
                break;
            }
            System.out.println("Tablero de juego:");
            System.out.println("\nCantidad de soldados en el Ejército 1: " + Soldado.getSEjer1());
            System.out.println("Cantidad de soldados en el Ejército 2: " + Soldado.getSEjer2());
            System.out.println("Cantidad de soldados en total: " + Soldado.getSoldCread());
            imprimirTablero(tablero, ejercito1, ejercito2);
            System.out.println("Turno del Jugador 2 - C ");
            moverSoldado(sc, tablero, ejercito2, ejercito1);
            if (ejercito1.isEmpty()) {
                System.out.println("¡El Jugador 2 ha ganado!");
                break;
            }
        }
    }
}
```

```
videojuego (1) [Java Application] C:\Users\USUARIO\Downloads\eclipse-java-2023
Menú:
1. Juego Rápido
2. Personalizado
3. Salir
Seleccione una opción: 1

Ejercito-1
Cantidad de soldados en el Ejército 1: 4
nombre: soldado0X1
fila: 2
columna: 1
vida: 4
nivel de Ataque: 1
nivel de Defensa: 5
velocidad: 5
nombre: soldado1X1
fila: 3
columna: 1
vida: 1
nivel de Ataque: 1
nivel de Defensa: 2
velocidad: 5
nombre: soldado2X1
fila: 5
columna: 2
vida: 2
nivel de Ataque: 1
nivel de Defensa: 1
velocidad: 5
nombre: soldado3X1
fila: 3
columna: 5
vida: 5
nivel de Ataque: 2
nivel de Defensa: 5
velocidad: 4
```

II. PRUEBA

III. CUESTIONARIO:

REPOSITORIO: https://github.com/UJKMjuandi/FundamentosP2

CONCLUSIONES

La expansión del programa para enfrentamientos entre jugadores humanos ha mejorado la experiencia del usuario mediante la introducción de métodos estratégicos. La selección de soldados, la planificación de movimientos y la gestión de turnos se han optimizado, resaltando la importancia de los atributos de la clase, como nivelAtaque y vive. La interfaz de usuario se ha mejorado, permitiendo una presentación clara de información. La gestión de eventos y resultados se ha afinado, utilizando atributos como vidaActual. La evaluación de fortalezas y debilidades se ha intensificado, destacando la relevancia de los atributos para estrategias tácticas, enriqueciendo así la experiencia de juego. En resumen, la sólida estructura de clase ha sido crucial para estas mejoras y para permitir una experiencia de juego más interactiva y estratégica.

METODOLOGÍA DE TRABAJO

1.- Leer los enunciados cuidadosamente para saber que requiere el problemas.

2.- verificar el diagrama uml puesto en la practica

3.-Empezar a hacer las clase soldado y la clase VideoJuego

4.- Realizar el Videojuego con los métodos apropiados.

REFERENCIAS Y BIBLIOGRAFÍA

[1] M. A. Lopez, E. Castro Gutierrez,*Fundamentos de la Programación 2 Topicos de Programación Orientada a Objetos*.Arequipa: UNSA,2021

RUBRICA PARA EL CONTENIDO DEL INFORME Y DEMOSTRACIÓN

El alumno debe marcar o dejar en blanco en celdas de la columna Checklist si cumplió con el ítem correspondiente.

Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos lo ítems.

El alumno debe autocalificarse en la columna Estudiante de acuerdo a la siguiente tabla:

Tabla 1: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	2	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega	2	X	0	

	establecidos.				
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	3	
TOTAL		20		13	

Tabla 2: Rúbrica para contenido del Informe y demostración