

	UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA	
Forma o: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 1
	UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA	
Forma o: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 2

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	Fundamentos de la Programacion 2				
TÍTULO DE LA PRÁCTICA:	<i>Laboratorio 11</i>				
NÚMERO DE PRÁCTICA:	<i>11</i>	AÑO LECTIVO:	<i>1</i>	NRO. SEMESTRE:	<i>2</i>
FECHA DE PRESENTACIÓN	<i>20/12/2023</i>	HORA DE PRESENTACIÓN	<i>4:35 pm</i>		
INTEGRANTE (s): Cuno Cahuari Armando Steven				NOTA:	
DOCENTE(s): Alain Revilla Christian					

SOLUCIÓN Y RESULTADOS
I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS <p>Es el mismo código del laboratorio pasado lo único que cambia es los valores de criterio que toma para sacar al ganador entre las batallas</p> <p>Se modifiko lo que viene a ser el método batalla el cual las métricas son mediante porcentajes y se decide al ganador de manera aleatoria.</p>

```

public static void batalla(String coordenada, int movimiento, ArrayList<Soldado> usuario, ArrayList<Soldado> co
    int fila_act, columna_act, fila_mov, columna_mov, posAct, us, cont, vidaAd;
    fila_act = Integer.parseInt(coordenada.substring(1, coordenada.length())) - 1;
    columna_act = nroColumna(coordenada.charAt(0));
    fila_mov = Movimiento.movFila(fila_act, movimiento);
    columna_mov = Movimiento.movColumna(columna_act, movimiento);
    String movRealizado = conversionBusqueda(columna_mov) + String.valueOf(fila_mov + 1);
    us = buscadorPosicion(usuario, coordenada);
    usuario.get(us).atacar();
    cont = buscadorPosicion(contrincante, movRealizado);
    contrincante.get(cont).defender();
    System.out.println("Jugador actual : " + usuario.get(us).getNivVidAct());
    System.out.println("Jugador enemigo : " + contrincante.get(cont).getNivVidAct());
    int total = usuario.get(us).getNivVidAct() + contrincante.get(cont).getNivVidAct();
    double probabilidad1 = usuario.get(us).getNivVidAct() * 100 / total;
    probabilidad1 = Math.round(probabilidad1 * 10.0) / 10.0;
    System.out.println("Posibilidad de porcentaje del Jugador actual: " + probabilidad1 + "%");
    System.out.println("Posibilidad de porcentaje del Jugador enemigo: " + (100 - probabilidad1) + "%");
    int nroRandom = (int)(Math.random() * 100 + 1);
    System.out.println("El numero aleatorio fue de " + nroRandom);
    if(0 <= nroRandom && nroRandom <= probabilidad1){
        contrincante.remove(cont);
        usuario.get(us).setFila(fila_mov + 1);
        usuario.get(us).setCol(conversionBusqueda(columna_mov));
        tablero[fila_mov][columna_mov] = usuario.get(us).getFigura();
        tablero[fila_act][columna_act] = '-';
        usuario.get(us).addVida();
    } else {
        usuario.remove(us);
        tablero[fila_act][columna_act] = '-';
        contrincante.get(cont).addVida();
    }
}

```

Con tal que saques el porcentaje de 1 como se muestra mediante probabilidad1 la variable doublé la cual es redondeada hacia un decimal asimismo como las métricas decían mediante un numero aleatorio para ello eso el valor de nroRandom el cual sacara aleatoriamente el valor.

Si en caso los limites desde 0 al valor de la 1ra probabilidad esta dentro ese valor aleatorio es ganador el jugador actual encaso contrario no. Por cada jugador que salga victorioso se le aumenta la vida mediante un addVida que viene siendo un método de instancia.

```

public void addVida(){
    vidaActual++;
}

```

Ejemplo:

Inicio de Juego

El tablero es :

```
-----  
-----  
&--&----&  
-----&--*  
--&*-----  
-&*-----  
--&--*---*  
-----*-----  
-----&-  
-----**--**
```

Turno del Ejercito 1

1 2 3

8 x 4

7 6 5

Ingrese otra coordenada

C6

6 2

Ingresar Movimiento:

2

4 Movimiento aplicado 2

Enemigo hallado

Jugador actual : 3

Jugador enemigo : 4

Posibilidad de porcentaje del Jugador actual: 42.0%

Posibilidad de porcentaje del Jugador enemigo: 58.0%

El numero aleatorio fue de 14

El tablero es :

```
-----  
-----  
&--&----&  
-----&--*  
--**-----  
-&-----  
--&--*---*  
-----*-----  
-----&-  
-----**--**
```

II. CONCLUSIONES

Podemos simplificar aun mas los códigos asimismo el uso que podemos aplicar mediante métodos para que no se vea tan desordenado asimismo mejorando la lógica de programación laboratorio tras laboratorio.

RETROALIMENTACIÓN GENERAL

Se debe mejorar la estructura del código para que no se vea tan aglomerado e desordenado asimismo mejorar los identificadores que estos poseen.

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 3</p>

REFERENCIAS Y BIBLIOGRAFÍA

--