


	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>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: 1</p>

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	Fundamentos de la Programación 2				
TÍTULO DE LA PRÁCTICA:	<i>Laboratorio 21 – Herencia y Polimorfismo.</i>				
NÚMERO DE PRÁCTICA:	21	AÑO LECTIVO:	1	NRO. SEMESTRE:	2
FECHA DE PRESENTACIÓN	24/01/2024	HORA DE PRESENTACIÓN	5:00 pm		
INTEGRANTE (s): Cuno Cahuari Armando Steven				NOTA:	

SOLUCIÓN Y RESULTADOS
<p>I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS</p> <p>Link GitHub: https://github.com/SteveArms/LaboratorioFPIL</p> <p>Código de Java:</p> <p>El inicio del programa se evalua de manera aleatoria el mapa.</p> <pre>//Metodo para definir el entorno del territorio public static String aleatorioMapa(){ int rand = (int)(Math.random() * 5 + 1); switch(rand){ case 1: return "BOSQUE"; case 2: return "CAMPO ABIERTO"; case 3: return "MONTANIA"; case 4: return "DESIERTO"; case 5: return "PLAYA"; default: return " "; } }</pre>

	<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: 2</p>

La implementación de lo que viene a ser los diferentes tipos de clases que puede ofrecernos los diferentes tipos de Soldados.

```
//La clase CaballeroFranco asimismo hereda atributos de la superclase Soldado la cual tiene otros metodos.
import java.util.*;
public class CaballeroFranco extends Soldado{
    private int cantidadLanzas;
    private int longitudLanza;
    private int evolucion = 1;
    public CaballeroFranco(String nombre, char figura){
        super(nombre, figura, vida:15);
        cantidadLanzas = (int)(Math.random() * 5 + 1);
        longitudLanza = (int)(Math.random() * 5 + 1);
    }
    public void evolucionarSoldado(){
        if(evolucion <= 4){
            if(cantidadLanzas < 5){
                cantidadLanzas++;
            } else if(longitudLanza < 5){
                longitudLanza++;
            }
        }
        evolucion++;
    }
}
```

Tenemos otra clase CaballeroMoro

```
public class CaballeroMoro extends Soldado{
    private int cantidadFlechas;
    private int longitudFlechas;
    private int evolucion = 1;
    public CaballeroMoro(String nombre, char figura){
        super(nombre, figura, vida:13);
        cantidadFlechas = (int)(Math.random() * 5 + 1);
        longitudFlechas = (int)(Math.random() * 5 + 1);
    }
    public void evolucionarSoldado(){
        if(evolucion <= 4){
            if(cantidadFlechas < 5){
                cantidadFlechas++;
            } else if(longitudFlechas < 5){
                longitudFlechas++;
            }
        }
        evolucion++;
    }
}
```

Tenemos otra clase EspadachinConquistador

```
public class EspadachinConquistador extends Soldado{
    private int cantidadFlechas;
    private int longitudFlechas;
    private int evolucion = 1;
    public EspadachinConquistador(String nombre, char figura){
        super(nombre, figura, vida:14);
        cantidadFlechas = (int)(Math.random() * 5 + 1);
        longitudFlechas = (int)(Math.random() * 5 + 1);
    }
    public void evolucionarSoldado(){
        if(evolucion <= 4){
            if(cantidadFlechas < 5){
                cantidadFlechas++;
            } else if(longitudFlechas < 5){
                longitudFlechas++;
            }
        }
        evolucion++;
    }
}
```

Tenemos otra clase EspadachinReal

```
import java.util.*;
public class EspadachinReal extends Soldado{
    private int cantCuchillos;
    private int longitudCuchillos;
    private boolean lanzarCuchillos;
    private int evolucion = 0;
    public EspadachinReal(String nombre, char figura){
        super(nombre, figura, vida:12);
        cantCuchillos = (int)(Math.random() * 5 + 1);
        longitudCuchillos = (int)(Math.random() * 5 + 1);
        lanzarCuchillos = true;
    }
    public void accionLanzarCuchillos(){
        if(cantCuchillos == 0){
            System.out.println(x:"Ya no quedan cuchillos");
        } else {
            cantCuchillos--;
        }
    }
    public void evolucionarSoldado(){
        if(evolucion <= 4){
            if(cantCuchillos < 5){
                cantCuchillos++;
            } else if(longitudCuchillos < 5){
                longitudCuchillos++;
            }
        }
        evolucion++;
    }
}
```

Tenemos la clase Teutocnico

```

public class EspadachinTeutocnico extends Soldado{
    private int numeroJabalinas;
    private int longitudJabalina;
    private int evolucion = 1;
    public EspadachinTeutocnico(String nombre, char figura){
        super(nombre, figura, vida:14);
        numeroJabalinas = (int)(Math.random() * 5 + 1);
        longitudJabalina = (int)(Math.random() * 5 + 1);
    }
    public void lanzar(){
        if(numeroJabalinas == 0){
            System.out.println(x:"No es posible que el espadachin lance jabalinas");
        } else {
            System.out.println(x:"El Espadachin lanzo una jabalina");
            numeroJabalinas--;
        }
    }
    public void evolucionarSoldado(){
        if(evolucion <= 4){
            if(numeroJabalinas < 5){
                numeroJabalinas++;
            } else if(longitudJabalina < 5){
                longitudJabalina++;
            }
        }
        evolucion++;
    }
}

```

Empleando los mismos métodos de Java para dar solución al problema para que este será el mismo trabajo en laboratorios anteriores en los cuales haremos uso de métodos que harán un reajuste en los valores asignados en las posiciones correspondientes por cada soldado viéndose esquemizado en un mapa el cual nos indicara sus opciones.



```

Scanner sc = new Scanner(System.in);
int opcion;
String nombre = "";
Ejercito actual, enemigo;
String territorio = aleatorioMapa();
Mapa tablero = new Mapa(territorio);
nombre = ingresarNombre();
actual = new Ejercito(nombre, fig:'*', numero:1);
nombre = ingresarNombre();
enemigo = new Ejercito(nombre, fig:'%', numero:2);
tablero.comprobarValores(actual.getSoldados());
tablero.comprobarValores(enemigo.getSoldados());

```

Asimismo, el método de juego empleado en cada batalla cuando se da el 1 vs 1 entre jugadores pues este después de cada jugada hará una impresión del mapa el cual nos dará un resumen de lo que viene a ser los ejércitos por cada lado.

Asimismo la ventaja que puede haber entre cada batalla viene a ser la implementación de agregación de vida que puede salir beneficioso según al tipo de Soldado correspondiente.



	<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: 5</p>

```

public static void agregarVidaBatalla(ArrayList<Soldado> usuario, ArrayList<Soldado> contrincante, int posAct, int posCont){
    Soldado prim = usuario.get(posAct);
    Soldado seg = contrincante.get(posCont);
    if((usuario.get(posAct) instanceof Caballero) && (contrincante.get(posCont) instanceof Arquero)){
        prim.addVida();
    } else if((usuario.get(posAct) instanceof CaballeroFranco) && (contrincante.get(posCont) instanceof Arquero)){
        prim.addVida();
        prim.addVida();
    } else if((usuario.get(posAct) instanceof CaballeroMoro) && (contrincante.get(posCont) instanceof Arquero)){
        prim.addVida();
        prim.addVida();
    } else if((usuario.get(posAct) instanceof Caballero) && (contrincante.get(posCont) instanceof Lancero)){
        seg.addVida();
    } else if((usuario.get(posAct) instanceof Arquero) && (contrincante.get(posCont) instanceof Lancero)){
        prim.addVida();
    } else if((usuario.get(posAct) instanceof Caballero) && (contrincante.get(posCont) instanceof Espadachin)){
        prim.addVida();
    } else if((usuario.get(posAct) instanceof Espadachin) && (contrincante.get(posCont) instanceof Lancero)){
        prim.addVida();
    } else if((usuario.get(posAct) instanceof EspadachinConquistador) && (contrincante.get(posCont) instanceof Lancero)){
        prim.addVida();
        prim.addVida();
    } else if((usuario.get(posAct) instanceof EspadachinReal) && (contrincante.get(posCont) instanceof Lancero)){
        prim.addVida();
        prim.addVida();
    } else if((usuario.get(posAct) instanceof EspadachinTeutocnico) && (contrincante.get(posCont) instanceof Lancero)){
        prim.addVida();
        prim.addVida();
    } else if((usuario.get(posAct) instanceof Espadachin) && (contrincante.get(posCont) instanceof EspadachinConquistador)){
        seg.addVida();
    } else if((usuario.get(posAct) instanceof Espadachin) && (contrincante.get(posCont) instanceof EspadachinReal)){
        seg.addVida();
    } else if((usuario.get(posAct) instanceof Espadachin) && (contrincante.get(posCont) instanceof EspadachinTeutocnico)){
        seg.addVida();
    } else if((usuario.get(posAct) instanceof Caballero) && (contrincante.get(posCont) instanceof CaballeroFranco)){
        seg.addVida();
    } else if((usuario.get(posAct) instanceof Caballero) && (contrincante.get(posCont) instanceof CaballeroMoro)){
        seg.addVida();
    }
}

```



El resumen expresado tras cada jugada al imprimir la batalla.

	<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: 6</p>

```
//Resumen que cuenta con indicacion del reino, cantidad de unidades, distribucion por unidades, nivel de vida total y porcentajes mas ganador.
public static void resumen(Ejercito actual, Ejercito enemigo){
    System.out.println("Ejercito 1: " + actual.getReino());
    System.out.println("Cantidad de Soldados creados: " + actual.cantidadSoldados());
    System.out.println("Espadachines : " + actual.getEspadachin());
    System.out.println("Arqueros : " + actual.getArquero());
    System.out.println("Caballerows : " + actual.getCaballero());
    System.out.println("Lanceros : " + actual.getLancero());
    System.out.println("Espadachin Real : " + actual.getEspadachinReal());
    System.out.println("Espadachin Conquistador : " + actual.getEspadachinConquistador());
    System.out.println("Espadachin Teutonico : " + actual.getEspadachinTeutonico());
    System.out.println("Caballero Franco : " + actual.getCaballeroFranco());
    System.out.println("Caballero Moro : " + actual.getCaballeroMoro());
    System.out.println();
    System.out.println("Ejercito 2: " + enemigo.getReino());
    System.out.println("Cantidad de Soldados creados: " + enemigo.cantidadSoldados());
    System.out.println("Espadachines : " + enemigo.getEspadachin());
    System.out.println("Arqueros : " + enemigo.getArquero());
    System.out.println("Caballerows : " + enemigo.getCaballero());
    System.out.println("Lanceros : " + enemigo.getLancero());
    System.out.println("Espadachin Real : " + enemigo.getEspadachinReal());
    System.out.println("Espadachin Conquistador : " + enemigo.getEspadachinConquistador());
    System.out.println("Espadachin Teutonico : " + enemigo.getEspadachinTeutonico());
    System.out.println("Caballero Franco : " + enemigo.getCaballeroFranco());
    System.out.println("Caballero Moro : " + enemigo.getCaballeroMoro());
    System.out.println();
    double total = actual.cantidadSoldados() + enemigo.cantidadSoldados();
    double prob = Math.round(((actual.cantidadSoldados() * 100.0) / total) * 100.0) / 100.0;
    double probabilidad = Math.round((Math.random() * 100 + 1) * 100.0) / 100.0;
    System.out.println("Ejercito 1 " + actual.getReino() + " : " + actual.cantidadSoldados() + " Probabilidad: " + prob);
    System.out.println("Ejercito 2 " + enemigo.getReino() + " : " + enemigo.cantidadSoldados() + " Probabilidad: " + (100 - prob));
    System.out.println("Por metricas el ganador es: El numero aleatorio fue " + probabilidad);
    if(probabilidad <= prob && probabilidad >= 0){
        System.out.println("Ganador el ejercito 1 " + actual.getReino());
    } else {
        System.out.println("Ganador el ejercito 2 " + enemigo.getReino());
    }
}
```

Ejecución de Código:

```
Ingrese el nombre del ejercito solo validos :
Inglaterra - Francia - Castilla o Aragon - Sacro - Moros
inglaterra
El ejercito tiene un total de 8 soldados
Ingrese el nombre del ejercito solo validos :
Inglaterra - Francia - Castilla o Aragon - Sacro - Moros
francia
El ejercito tiene un total de 7 soldados
-- -- -- -- --
L% L* -- -- -- L* -- --
L* -- -- -- A* -- -- -- A* --
-- -- -- L* -- -- --
E% -- -- -- A% -- -- --
-- -- -- -- --
-- -- -- A% -- -- -- C*
-- -- -- E% L* -- -- --
-- -- -- -- --
Turno del Ejercito *
1 2 3
8 x 4
7 6 5
Ingrese otra coordenada
A4
4
0
Coordenada incorrecta
1 2 3
8 x 4
7 6 5
Ingrese otra coordenada
B2
2
1
Ingresar Movimiento:
8
1 Movimiento aplicado 0
Enemigo hallado
Jugador actual : 6
Jugador enemigo : 5
Posibilidad de porcentaje del Jugador actual: 54.0%
Posibilidad de porcentaje del Jugador enemigo: 46.0%
```

	<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: 7</p>

```

Posibilidad de porcentaje del Jugador enemigo: 46.0%
El numero aleatorio fue de 85.92
-- -- -- -- --
L% -- -- -- -- -- L* -- -- -- -- --
L* -- -- -- -- -- A* -- -- -- -- -- A* --
-- -- -- -- -- L* -- -- -- -- --
E% -- -- -- -- -- A% -- -- -- -- --
-- -- -- -- -- A% -- -- -- -- -- C*
-- -- -- -- -- E% L* -- -- -- -- --
-- -- -- -- --
Desea seguir jugando? Escriba no si no desea
no
Salio victorioso el ejercito 1
Nombre: 1C2 Vida:15 Fila:6 Ataque:4 Defensa:4 Columna:I
Arquero Nombre: 2A2 Vida:3 Fila:5 Ataque:7 Defensa:3 Columna:E
Arquero Nombre: 3A2 Vida:5 Fila:7 Ataque:7 Defensa:3 Columna:F
Nombre: 4C2 Vida:15 Fila:4 Ataque:3 Defensa:4 Columna:D
Espadachin Nombre: 5E2 Vida:9 Fila:5 Ataque:10 Defensa:8 Columna:A
Lancero Nombre: 6L2 Vida:6 Fila:2 Ataque:5 Defensa:10 Columna:A
Espadachin Nombre: 7E2 Vida:10 Fila:8 Ataque:10 Defensa:8 Columna:D
Ejercito 1: INGLATERRA
Cantidad de Soldados creados: 7
Espadachines : 0
Arqueros : 2
Caballerows : 1
Lanceros : 5
Espadachin Real : 0
Espadachin Conquistador : 0
Espadachin Teutonico : 0
Caballero Franco : 0
Caballero Moro : 0



Ejercito 2: FRANCIA
Cantidad de Soldados creados: 7
Espadachines : 0
Arqueros : 2
Caballerows : 2
Lanceros : 1
Espadachin Real : 0
Espadachin Conquistador : 0
Espadachin Teutonico : 0
Caballero Franco : 2
Caballero Moro : 0

Ejercito 1 INGLATERRA : 7 Probabilidad: 50.0
Ejercito 2 FRANCIA : 7 Probabilidad: 50.0
Por metricas el ganador es: El numero aleatorio fue 37.98
Ganador el ejercito 1 INGLATERRA

```

II. CONCLUSIONES

Las conclusiones dadas es que la implementación de métodos, miembros de clase que se dan por clases de Objeto pueden beneficiar de manera global a la redacción de programación pueden convenir mucho a lo que viene a ser el lenguaje de programación.

	<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: 8</p>

RETROALIMENTACIÓN GENERAL

Debemos mejorar el empleo de lo que viene a ser los métodos necesarios para su conveniencia lo cual mejorara la sintaxis y el entendimiento de igual forma la aplicación y los métodos que pueden ofrecer cada diferente tipo de clase.

REFERENCIAS Y BIBLIOGRAFÍA

Fundamentos de la Programacion2. Tópicos de Programación orientada a objetos por Marcos Aedo y Eveling Castro
Java: Conceptos básicos de programación para principiantes por Nathan Clark