
	<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: 1</p>

INFORME DE LABORATORIO

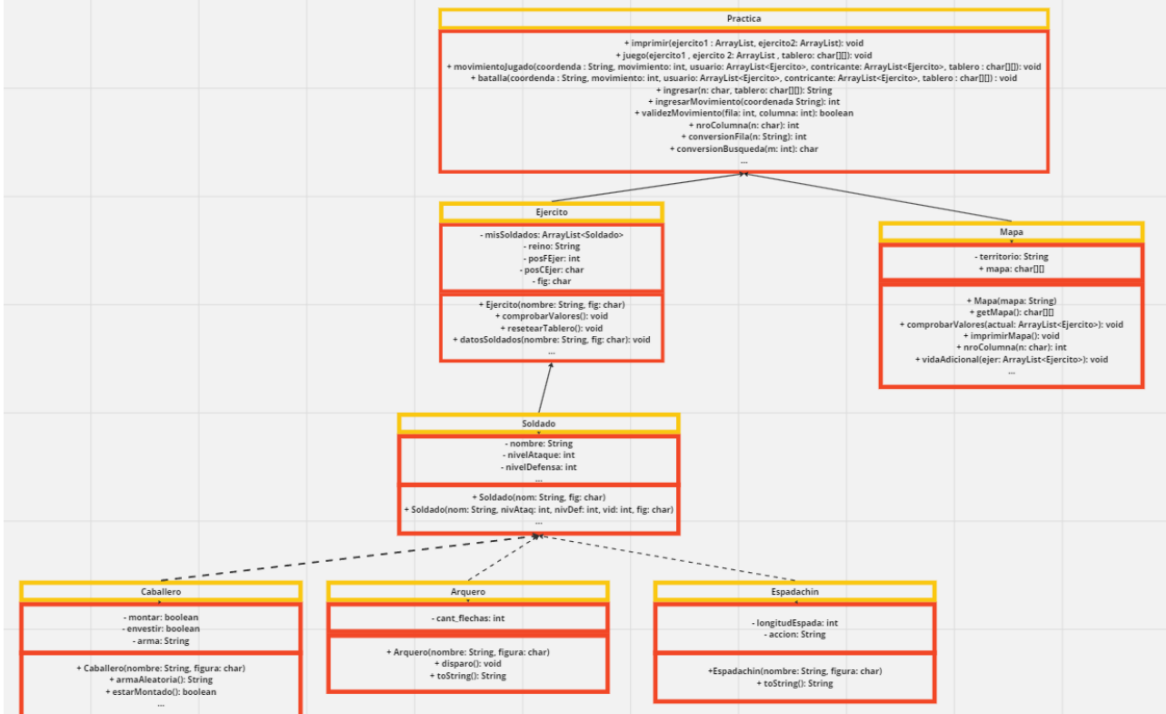
INFORMACIÓN BÁSICA					
ASIGNATURA:	Fundamentos de la Programación 2				
TÍTULO DE LA PRÁCTICA:	<i>Laboratorio 22 – Interfaz gráfica de usuario</i>				
NÚMERO DE PRÁCTICA:	22	AÑO LECTIVO:	1	NRO. SEMESTRE:	2
FECHA DE PRESENTACIÓN	24/01/2024	HORA DE PRESENTACIÓN	10:30 am		
INTEGRANTE (s): Cuno Cahuari Armando Steven				NOTA:	

SOLUCIÓN Y RESULTADOS

I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS

Link GitHub:
<https://github.com/SteveArms/LaboratorioFPIL.git>

Diagrama UML:



```

classDiagram
    class Practica {
        +imprimirEjercito1 : ArrayList, ejercito2: ArrayList: void
        +juegoEjercito1, ejercito2: ArrayList, tablero: char[][]: void
        +movimientoJugador(coordenda : String, movimiento: int, usuario: ArrayList-Ejercito, contricante: ArrayList-Ejercito, tablero : char[][]: void
        +batalla(coordenda : String, movimiento: int, usuario: ArrayList-Ejercito, contricante: ArrayList-Ejercito, tablero: char[][]: void
        +ingresarEjercito: char, tablero: char[][]: String
        +ingresarMovimiento(coordenda String): int
        +validarMovimiento(fila: int, columna: int): boolean
        +nroColumnas: char: int
        +conversionFila(n: String): int
        +conversionBusqueda(m: int): char
    }
    class Ejercito {
        -misSoldados: ArrayList-Soldado
        -reino: String
        -posEjer: int
        -posEjer: char
        -fig: char
        +Ejercito(nombre: String, fig: char)
        +comprobarValores(): void
        +resetearTablero(): void
        +datosSoldados(nombre: String, fig: char): void
    }
    class Mapa {
        -territorio: String
        +mapa: char[][]
        +Mapa(mapa: String)
        +getMapa(): char[][]
        +comprobarValores(actual: ArrayList-Ejercito): void
        +imprimirMapa(): void
        +nroColumnas(n: char): int
        +vidaAdicionalEjer: ArrayList-Ejercito): void
    }
    class Soldado {
        -nombre: String
        -nivelAtaque: int
        -nivelDefensa: int
        +Soldado(nom: String, fig: char)
        +Soldado(nom: String, nivAtaq: int, nivDef: int, vid: int, fig: char)
    }
    class Caballero {
        -montar: boolean
        -envestir: boolean
        -arma: String
        +Caballero(nombre: String, figura: char)
        +armaAleatoria(): String
        +estarMontado(): boolean
    }
    class Arquero {
        -cant, flechas: int
        +Arquero(nombre: String, figura: char)
        +disparo(): void
        +toString(): String
    }
    class Espadachin {
        -longitudEspada: int
        -accion: String
        +Espadachin(nombre: String, figura: char)
        +toString(): String
    }
    Practica --> Ejercito
    Practica --> Mapa
    Ejercito --> Soldado
    Soldado ..> Caballero
    Soldado ..> Arquero
    Soldado ..> Espadachin
  
```

Código:

Acerca de la clase interfaz grafica del usuario se ve aplicada en diversos botones que aplican variedad de funciones haciendo realización de diversas actividades al hacer uso de diversos botones.

Los métodos ofrecidos para el evento cuando es activado ofreciendo diversas actividades entre ellos. Estos métodos que ofrecerán funciones diferentes son aquellos que fueron trabajados en laboratorios anteriores. Estos métodos vienen a ser de instancia al hacer creación del objeto Ejercito.

```
//Metodo necesario para imprimir que tipo de Soldado es cada uno segun en su posicion correspondiente
public static void mapaSoldados(ArrayList<Soldado> actual, ArrayList<Soldado> enemigo){
    String[][] mapa = {{"-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-"},
        {"-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-"},
        {"-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-"},
        {"-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-"},
        {"-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-"},
        {"-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-"},
        {"-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-"},
        {"-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-"},
        {"-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-"},
        {"-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-"}};

    for(Soldado m: actual){
        if(m instanceof Arquero){
            mapa[m.getFila() - 1][m.getColumna()] = "A" + m.getFigura();
        } else if(m instanceof Caballero){
            mapa[m.getFila() - 1][m.getColumna()] = "C" + m.getFigura();
        } else if(m instanceof Espadachin){
            mapa[m.getFila() - 1][m.getColumna()] = "E" + m.getFigura();
        } else if(m instanceof Lancero){
            mapa[m.getFila() - 1][m.getColumna()] = "L" + m.getFigura();
        }
    }

    for(Soldado n: enemigo){
        if(n instanceof Arquero){
            mapa[n.getFila() - 1][n.getColumna()] = "A" + n.getFigura();
        } else if(n instanceof Caballero){
            mapa[n.getFila() - 1][n.getColumna()] = "C" + n.getFigura();
        } else if(n instanceof Espadachin){
            mapa[n.getFila() - 1][n.getColumna()] = "E" + n.getFigura();
        } else if(n instanceof Lancero){
            mapa[n.getFila() - 1][n.getColumna()] = "L" + n.getFigura();
        }
    }

    for(int i = 0; i < mapa.length; i++){
        for(int j = 0; j < mapa[i].length; j++){
            System.out.print(mapa[i][j] + " ");
        }
        System.out.println();
    }
}

//Metodo para el promedio de vida mediante la ayuda de otros metodos de instancia
public void promedioVida(){
    System.out.println("El promedio de vida del ejercito " + reino + " es: ");
    System.out.println(this.cantidadVida() / this.cantidadSoldados());
}
```

```
//Metodo Soldados con mayor vida necesario hacer una busqueda lineal con la mayor cantidad de vida e imprimir todos los soldados que tengan ese nivel de vida
public void soldadosMayorVida(){
    int mayor = misSoldados.get(index:0).getVida();
    for(int i = 1; i < misSoldados.size(); i++){
        if(mayor < misSoldados.get(i).getVida()){
            mayor = misSoldados.get(i).getVida();
        }
    }
    System.out.println("Los soldados con mayor vida del ejercito " + reino + " son :");
    for(int i = 0; i < misSoldados.size(); i++){
        if(mayor == misSoldados.get(i).getVida()){
            System.out.println(misSoldados.get(i));
        }
    }
}

public void imprimir(){
    System.out.println("Ejercito " + reino);
    for(Soldado m: misSoldados){
        System.out.println(m);
    }
}

//Resumen que cuenta con indicacion del reino, cantidad de unidades, distribucion por unidades, nivel de vida total y porcentajes mas ganador.
public static String resumen(Ejercito actual, Ejercito enemigo){
    return "Ejercito 1: " + actual.getReino() + "\n" +
        "Cantidad de Soldados creados: " + actual.cantidadSoldados() + "\n" +
        "Espadachines : " + actual.getEspadachin() + "\n" +
        "Arqueros : " + actual.getArquero() + "\n" +
        "Caballeros : " + actual.getCaballero() + "\n" +
        "Lanceros : " + actual.getLancero() + "\n\n" +
        "Ejercito 2: " + enemigo.getReino() + "\n" +
        "Cantidad de Soldados creados: " + enemigo.cantidadSoldados() + "\n" +
        "Espadachines : " + enemigo.getEspadachin() + "\n" +
        "Arqueros : " + enemigo.getArquero() + "\n" +
        "Caballeros : " + enemigo.getCaballero() + "\n" +
        "Lanceros : " + enemigo.getLancero() + "\n\n";
}
}
```



Estos métodos se verán activados mediante eventos los cuales funcionarán por botones. Que harán activar diferente método que se visualizara mediante la consola asimismo al finalizar el programa este mandara una ventana emergente informando que está finalizando el programa.

Crearemos un objeto JFrame llamado ventanaBienvenida los cuales se modificarán mediante métodos de instancia los cuales modificaras la forma que tiene la ventana.

```
//Implementamos los objetos que pueden ser usados junto al JFrame tanto como botones y Label
JFrame ventanaBienvenida = new JFrame(title:"Bienvenido al Juego");
ventanaBienvenida.setSize(width:300, height:150);
ventanaBienvenida.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

ventanaBienvenida.add(panelPrincipal);
ventanaBienvenida.setLocationRelativeTo(c:null);
ventanaBienvenida.setVisible(b:true);
```

Asimismo, los complementos correspondientes que tendrá esta ventana pues añadiremos lo que viene a ser objetos los cuales harán interactivo y vistoso a nuestro JFrame modificando mediante propiedades asimismo el color que tendrán estos objetos que lo alberguen.

	<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: 4</p>

```
//Instanciamos todos los objetos que complementaran al JFrame
JPanel panelPrincipal = new JPanel(new FlowLayout(FlowLayout.CENTER, hgap:10, vgap:30));
JLabel mensajeLabel = new JLabel(text:"----- Menu de Opciones ----- ");
JButton boton1ra = new JButton(text:" 1ra Opcion ");
JButton boton2da = new JButton(text:" 2da Opcion");
JButton boton3ro = new JButton(text:" 3ra Opcion ");
JButton boton4to = new JButton(text:" 4ta Opcion ");
JButton boton5to = new JButton(text:" 5ta Opcion ");
JButton botonFinalizar = new JButton(text:" Finalizar ");
//Uso de metodos de instancia en los botones
boton1ra.setBackground(Color.GREEN);
boton2da.setBackground(Color.RED);
boton3ro.setBackground(Color.GRAY);
boton4to.setBackground(Color.MAGENTA);
boton5to.setBackground(Color.CYAN);
botonFinalizar.setBackground(Color.BLUE);

panelPrincipal.add(mensajeLabel);
panelPrincipal.add(boton1ra);
panelPrincipal.add(boton2da);
panelPrincipal.add(boton3ro);
panelPrincipal.add(boton4to);
panelPrincipal.add(boton5to);
panelPrincipal.add(botonFinalizar);
```

Acá instanciamos lo que viene a ser el JPanel , JLabel y los JButton que serán instanciados añadiendo asimismo el color de este. Los JButton serán agregados a lo que viene a ser el JPanel albergando todos los valores que serán instanciados.



Ahora asimismo lo que viene a ser los eventos que vayan a suceder por la interacción mediante un click en el cual cada botón ofrecerá diferente evento.

```
//Los eventos sucedidos al interactuar con Los botones pues estos ejecutarn diferentes funciones
//El action performed encargado de manejar el evento de la accion
//El actio event encargado de producir el evento.
boton1ra.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        tablero.mapaSoldados(actual.getSoldados(), enemigo.getSoldados());
    }
});
boton2da.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.out.println(x:"*****");
        actual.promedioVida();
        enemigo.promedioVida();
        System.out.println(x:"*****");
    }
});
boton3ro.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e){
        System.out.println(x:"*****");
        actual.soldadosMayorVida();
        enemigo.soldadosMayorVida();
        System.out.println(x:"*****");
    }
});
boton4to.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e){
        System.out.println(x:"*****");
        actual.imprimir();
        System.out.println(x:"*****");
        enemigo.imprimir();
        System.out.println(x:"*****");
    }
});
boton5to.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e){
        JOptionPane.showMessageDialog(parentComponent:null, resumen(actual, enemigo));
    }
});
botonFinalizar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e){
        ventanaBienvenida.dispose();
        JOptionPane.showMessageDialog(parentComponent:null, message:"Programa Finalizado");
    }
});
});
```

Una vez terminado las interacciones y los eventos que vayan a suceder mediante los botones pues estos se agregan al JPanel y este al finalizar, una vez albergado los botones será agregado a lo que viene a ser la ventana JFrame dando por finalizado el objeto JFrame.

```
ventanaBienvenida.add(panelPrincipal);
ventanaBienvenida.setLocationRelativeTo(c:null);
ventanaBienvenida.setVisible(b:true);
```

Ejecución del código:

	<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>



```

Ingrese el nombre del 1er Reino
Ingrese el nombre del ejercito solo validos :
Inglaterra - Francia - Castilla o Aragon - Sacro - Moros
francia
El ejercito tiene un total de 3 soldados
Ingrese el nombre del 2do reino
Ingrese el nombre del ejercito solo validos :
Inglaterra - Francia - Castilla o Aragon - Sacro - Moros
inglaterra
El ejercito tiene un total de 5 soldados
-- E* -- -- A* -- -- -- A% E%
-- -- -- C* -- -- -- C% --
-- -- -- -- -- -- -- --
-- -- -- -- -- -- -- --
-- -- -- -- -- -- -- --
-- -- A% -- -- -- -- -- --
-- -- -- -- -- -- A% -- -- --
-- -- -- -- -- -- -- --
-- -- -- -- -- -- -- --
-- -- -- -- -- -- -- --
*****
El promedio de vida del ejercito FRANCIA es:
8
El promedio de vida del ejercito INGLATERRA es:
6
*****
*****
Los soldados con mayor vida del ejercito FRANCIA son :
Caballero Nombre: 1C1 Vida:12 Fila:2 Ataque:13 Defensa:7 Columna:D
Los soldados con mayor vida del ejercito INGLATERRA son :
Caballero Nombre: 3C2 Vida:12 Fila:2 Ataque:13 Defensa:7 Columna:I
*****
*****
Ejercito FRANCIA
Caballero Nombre: 1C1 Vida:12 Fila:2 Ataque:13 Defensa:7 Columna:D
Espadachin Nombre: 2E1 Vida:9 Fila:1 Ataque:10 Defensa:8 Columna:B
Arquero Nombre: 3A1 Vida:3 Fila:1 Ataque:7 Defensa:3 Columna:E
*****
Ejercito INGLATERRA
Arquero Nombre: 1A2 Vida:5 Fila:6 Ataque:7 Defensa:3 Columna:C
Arquero Nombre: 2A2 Vida:4 Fila:1 Ataque:7 Defensa:3 Columna:I
Caballero Nombre: 3C2 Vida:12 Fila:2 Ataque:13 Defensa:7 Columna:I
Espadachin Nombre: 4E2 Vida:9 Fila:1 Ataque:10 Defensa:8 Columna:J
Arquero Nombre: 5A2 Vida:4 Fila:7 Ataque:7 Defensa:3 Columna:G
*****

```

II. CONCLUSIONES

Las opciones expresadas por el mismo lenguaje de programación java para la ejecución de interfaces graficas que hacen mas interactivo y atractivo al diseño y objetos de programación que puede ofrecernos java en toda su expresión. Asimismo, los objetos que complementan al objeto JFrame hacen que posea mas interactivo el programa hacia el público.

	<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: 7</p>

RETROALIMENTACIÓN GENERAL

Hubiera sido conveniente que al ejecutar los métodos mediante las funciones de las ventanas este se vea visualizado en otra ventana la cual se sobrescriba el código por encima demostrando la función diferente que tiene por demás de las otras.

REFERENCIAS Y BIBLIOGRAFÍA

Fundamentos de la Programación 2. Tópicos de Programación orientada a objetos por Marcos Aedo y Eveling Castro
Java: Conceptos básicos de programación para principiantes por Nathan Clark.