

Informe de Laboratorio 22

Tema: Interfaz Gráfica de Usuario

Nota

Estudiante	Escuela	Asignatura
Roni Companocca Checco rcompanocca@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Semestre: II Código: 20210558

Laboratorio	Tema	Duración
22	Interfaz Gráfica de Usuario	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 14 Diciembre 2023	Al 20 Diciembre 2023

1. TAREA

1.1. Objetivos:

- Crear componentes gráficos básicos
- Utilizar la interfaz ActionListener
- Crear inner classes

1.2. Competencias a alcanzar:

- Diseña, responsablemente, sistemas, componentes o procesos para satisfacer necesidades dentro de restricciones realistas: económicas, medio ambientales, sociales, políticas, éticas, de salud, de seguridad, manufacturación y sostenibilidad.
- Aplica de forma flexible, técnicas, métodos, principios, normas, estándares y herramientas de ingeniería necesarias para la construcción de software e implementación de sistemas de información.

2. EQUIPOS, MATERIALES Y TEMAS UTILIZADOS

- Sistema Operativo Windows
- OpenJDK 64-Bits 17.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.

3. URL DE REPOSITORIO GITHUB

- URL para el Repositorio GitHub.
- <https://github.com/RONI-COMPANOECA-CHECCO>
- URL para el laboratorio 22 en el Repositorio GitHub.
- <https://github.com/RONI-COMPANOECA-CHECCO/FP2-LAB22>

4. CODIGO DEL VIDEOJUEGO DE ESTRATEGIA Y SU DIAGRAMA UML

- clase Arquero.java

```
public class Arquero extends Soldado {  
    private int numeroFlechas;  
    private static int num = 0;  
  
    public Arquero(String s, int ata, int def, int vid, int cant){  
        super(s,ata,def,vid);  
        numeroFlechas = cant;  
        num++;  
    }  
  
    public void disparar(){  
        if(numeroFlechas>0){  
            numeroFlechas--;  
        }  
    }  
  
    public static int cuantos(){  
        return num;  
    }  
  
    public static void resetearCantidad(){  
        num = 0;  
    }  
  
    public String toString(){  
        return super.toString()+" "+numeroFlechas;  
    }  
}
```

- clase Caballero.java

```
public class Caballero extends Soldado{  
    private String armaActual = "lanza";  
    private boolean montando = true;  
    private static int num = 0;  
  
    public Caballero(String s, int ata, int def, int vid){  
        super(s,ata,def,vid);  
        num++;  
    }  
}
```

```
public void investir(){
    if(montando==true){
        for(int i=0; i<=2; i++){
            super.atacar();
        }
    }else{
        super.atacar();
    }
}

public void desmontar(){
    if(montando==true){
        montando = false;
        super.defender();
        cambiaArma();
    }
}

public void cambiaArma(){
    if(armaActual=="lanza"){
        armaActual = "Espada";
    }else{
        armaActual = "lanza";
    }
}

public void montar(){
    if(montando==false){
        montando = true;
        super.atacar();
        cambiaArma();
    }
}

public static int cuantos(){
    return num;
}

public static void resetearCantidad(){
    num=0;
}

public String toString(){
    return super.toString()+" "+armaActual+" "+montando;
}
}
```

■ clase Espadachin.java

```
public class Espadachin extends Soldado{
    private int longitudEspada;
    private boolean muroEscudos = false;
    private static int num = 0;

    public Espadachin(String s, int ata, int def, int vid, int lon){
```

```
        super(s,ata,def,vid);
        longitudEspada = lon;
        num++;
    }

    public void muroEscudos(){
        if(muroEscudos == true ){
            muroEscudos = false;
        }else{
            muroEscudos = true;
        }
    }

    public static int cuantos(){
        return num;
    }

    public static void resetearCantidad(){
        num =0;
    }

    public String toString(){
        return super.toString()+" "+longitudEspada+" "+muroEscudos;
    }
}
```

■ clase Lancero.java

```
public class Lancero extends Soldado {
    private int longitudLanza;
    private static int num = 0;

    public Lancero(String nombre, int ataque, int defensa, int vida, int longitudLanza)
    {
        super(nombre, ataque, defensa, vida);
        this.longitudLanza = longitudLanza;
        num++;
    }

    // Mtodo especifico para la accin "schiltrom"
    public void schiltrom() {
        if (getActitud().equals("defensa")) {
            // Aumentar el nivel de defensa en 1 cuando se realiza schiltrom
            setNivelDefensa(getNivelDefensa() + 1);
        }
    }

    // Sobrescribe el mtodo serAtacado para personalizar la accin en caso de ataque
    @Override
    public void serAtacado() {
        // Realiza schiltrom antes de recibir el ataque
        schiltrom();
        // Llama al mtodo de la superclase para aplicar el dao
        super.serAtacado();
    }
}
```

```
public static int cuantos(){
    return num;
}

public static void resetearCantidad(){
    num =0;
}

// Sobrescribe el mtodo toString para agregar informacin especifica de Lancero
@Override
public String toString() {
    return super.toString() + " Longitud de lanza: " + longitudLanza;
}
}
```

■ clase Ejercito.java

```
import java.util.*;
public class Ejercito {
    ArrayList<Soldado> misSoldados = new ArrayList<Soldado>();
    String cultura;

    public Ejercito(String cult, int cantidad){
        cultura = cult;
        int tipo;
        for(int i=0; i<cantidad; i++){
            tipo = (int)(Math.random()*4)+1;
            switch (tipo) {
                case 1: misSoldados.add(new Espadachin("\nE: "+i, 10, 8, 10, 40));
                    break;

                case 2: misSoldados.add(new Caballero("\nC: "+i, 13, 7, 12));
                    break;

                case 3: misSoldados.add(new Arquero("\nA: "+i, 7, 3, 5, 20));
                    break;

                case 4: misSoldados.add(new Lancero("\nL: "+i, 5, 10, 8, 20));
                    break;
            }
        }
    }

    public String toString(){
        String todos ="";
        for(int i=0; i<misSoldados.size(); i++){
            todos += misSoldados.get(i)+" ";
        }
        return cultura+" "+misSoldados.size()+" "+todos;
    }

    public int poder(){
        int poder = 0;
        for(int i=0; i<misSoldados.size(); i++){
            poder += misSoldados.get(i).getVida();
        }
    }
}
```

```
        return poder;
    }

    public String getCultura(){
        return cultura;
    }

    public static void resetearCantidad(){
        Soldado.resetearCantidad();
        Arquero.resetearCantidad();
        Caballero.resetearCantidad();
        Espadachin.resetearCantidad();
        Lancero.resetearCantidad();
    }
}
```

■ clase Mapa.java

```
import java.util.Random;

public class Mapa {
    private String tipoTerritorio;
    private Soldado[][] tablero;

    // Constructor
    public Mapa(String tipoTerritorio) {
        this.tipoTerritorio = tipoTerritorio;
        this.tablero = new Soldado[10][10]; // Tamao del tablero (puedes ajustarlo segn
        tus necesidades)
        posicionarSoldadosAleatorios();
    }

    // Mtodo para posicionar soldados aleatorios en el tablero
    private void posicionarSoldadosAleatorios() {
        Random rand = new Random();
        for (int i = 0; i < 10; i++) { // Nmero arbitrario de soldados por ejrcito
            int fila = rand.nextInt(10);
            int columna = rand.nextInt(10);

            // Verificar si la posicin est ocupada
            while (tablero[fila][columna] != null) {
                fila = rand.nextInt(10);
                columna = rand.nextInt(10);
            }

            // Crear un soldado aleatorio (puedes ajustar estos valores segn tus
            necesidades)
            Soldado soldado = new Soldado("Soldado" + i, 8, 5, rand.nextInt(5) + 5);
            tablero[fila][columna] = soldado;
        }
    }

    // Mtodo para mostrar el tablero
    public void mostrarTablero() {
        System.out.println("Mapa - Tipo de Territorio: " + tipoTerritorio);
        for (Soldado[] fila : tablero) {
```

```
        for (Soldado soldado : fila) {
            if (soldado != null) {
                System.out.print(soldado.getNombre() + " ");
            } else {
                System.out.print("Vaco ");
            }
        }
        System.out.println();
    }
}
```

■ clase Soldado.java

```
public class Soldado {
    private String nombre;
    private int nivelAtaque;
    private int nivelDefensa;
    private int nivelVida;
    private int vidaActual;
    private int velocidad = 0;
    private String actitud = "defensa";
    private boolean vive = true;
    private static int num = 0;

    //COMSTRUCTORES
    public Soldado(String nomb, int ataque, int defensa, int vida) {
        nombre = nomb;
        nivelAtaque = ataque;
        nivelDefensa = defensa;
        nivelVida = vida;
        vidaActual = vida;
        num++;
    }

    // Otros mtodos
    public void atacar() {
        actitud = "ataque";
        avanzar();
    }

    public void defender() {
        actitud = "defensa";
        velocidad = 0;
    }

    public void avanzar() {
        velocidad++;
    }

    public void retroceder() {
        velocidad--;
    }

    public void serAtacado() {
        vidaActual--;
    }
}
```

```
        if (vidaActual == 0)
            morir();
    }

    public void huir(){
        actitud = "fuga";
        velocidad++;
    }

    public void morir() {
        vive = false;
    }

    public String toString() {
        return nombre+ " " +nivelAtaque+ " "+nivelDefensa+ " " +nivelVida+ " "
            +vidaActual+ " " +velocidad+ " " +actitud+ " " +vive;
    }

    public static int cuantos(){
        return num;
    }

    public static void resetearCantidad(){
        num=0;
    }

    public int getVida(){
        return nivelVida;
    }

    public String getActitud(){
        return actitud;
    }

    public void setNivelDefensa(int nivelDefensa) {
        this.nivelDefensa = nivelDefensa;
    }

    public int getNivelDefensa(){
        return nivelDefensa;
    }
}
```

■ clase Main.java

```
// RONI COMPANOCCA CHECCO
// CUI: 20210558
// LABORATORIO 22
// FUNDAMENTOS DE PROGRAMACION
public class Main {
    public static void main(String[] args){
        int cant;
        String cultura[] = {"Inglaterra","Francia","Sacro Imperio Romano
            Germanico","Aragon","Moros"};
        cant = aleatorio(1,10);
        Ejercito e1 = new Ejercito(cultura[aleatorio(0,4)], cant);
    }
}
```



```
        mostrar(e1);
        cant = aleatorio(1,10);
        Ejercito e2 = new Ejercito(cultura[aleatorio(0,4)], cant);
        mostrar(e2);
        determinarGanador(e1,e2);
    }

    public static int aleatorio(int a, int b){
        return (int)(Math.random()*(b-a+1))+a;
    }

    private static int aux = 1;

    public static void mostrar(Ejercito e){
        System.out.print("Ejercito "+aux+" "+e.getCultura());
        System.out.println("\nCantidad total de Soldados:
            "+Soldado.cuantos()+"\n"+"Espadachines:
            "+Espadachin.cuantos()+"\n"+"Arqueros: "+Arquero.cuantos()+"\n"+"Lanceros:
            "+Lancero.cuantos()+"\n"+"Caballeros: "+Caballero.cuantos()+"\n");
        Ejercito.resetearCantidad();
        aux++;
    }

    public static void determinarGanador(Ejercito e1, Ejercito e2){
        System.out.println("Ejercito 1: "+e1.getCultura()+" con un poder de
            "+e1.poder());
        System.out.println("Ejercito 2: "+e2.getCultura()+" con un poder de
            "+e2.poder());
        if (e1.poder()>e2.poder()){
            System.out.println("El ganador es ejercito 1 de : "+e1.getCultura());
        } else if (e1.poder()<e2.poder()){
            System.out.println("El ganador es ejercito 2 de : "+e2.getCultura());
        } else{
            System.out.println("Sin ganador");
        }
    }
}
```



5. ACTIVIDAD

- Cree una versión del videojuego de estrategia usando componentes básicos GUI: Etiquetas, botones, cuadros de texto, JOptionPane, Color.
- Además, utilizar componentes avanzados GUI: Layouts, JPanel, áreas de texto, checkbox, botones de radio y combobox.
- Considerar nivel estratégico y táctico.
- Considerar hasta las unidades especiales de los reinos.
- Hacerlo iterativo.

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;

public class JuegoInterfazAvanzada extends JFrame {

    private JButton botonIniciar;

```

```
private JLabel etiquetaCultura;
private JComboBox<String> listaCulturas;
private JCheckBox checkBoxModoEstrategico;
private JRadioButton radioTactico;
private JRadioButton radioEstrategico;
private ButtonGroup grupoRadio;
private JTextArea areaTexto;
private JPanel panelJuego;

public JuegoInterfazAvanzada() {
    super("Juego de Estrategia");

    etiquetaCultura = new JLabel("Seleccione Territorio:");
    listaCulturas = new JComboBox<>(new String[]{"Bosque", "Campo abierto",
        "Montaa", "Desierto", "Playa"});
    checkBoxModoEstrategico = new JCheckBox("Modo Estratgico");

    radioTactico = new JRadioButton("Tctico");
    radioEstrategico = new JRadioButton("Estratgico");
    grupoRadio = new ButtonGroup();
    grupoRadio.add(radioTactico);
    grupoRadio.add(radioEstrategico);

    areaTexto = new JTextArea(10, 30);
    areaTexto.setEditable(false);
    JScrollPane scrollPane = new JScrollPane(areaTexto);

    botonIniciar = new JButton("Iniciar Juego");
    botonIniciar.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            iniciarJuego();
        }
    });

    panelJuego = new JPanel(new BorderLayout());
    add(panelJuego, BorderLayout.CENTER);

    JPanel panelSuperior = new JPanel(new FlowLayout());
    panelSuperior.add(etiquetaCultura);
    panelSuperior.add(listaCulturas);
    panelSuperior.add(checkBoxModoEstrategico);

    JPanel panelRadio = new JPanel();
    panelRadio.add(radioTactico);
    panelRadio.add(radioEstrategico);

    JPanel panelCentral = new JPanel(new BorderLayout());
    panelCentral.add(panelSuperior, BorderLayout.NORTH);
    panelCentral.add(panelRadio, BorderLayout.CENTER);
    panelCentral.add(scrollPane, BorderLayout.SOUTH);

    JPanel panelBoton = new JPanel();
    panelBoton.add(botonIniciar);

    add(panelCentral, BorderLayout.CENTER);
}
```

```
add(panelBoton, BorderLayout.SOUTH);

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
pack();
setLocationRelativeTo(null);
setVisible(true);
}

private void iniciarJuego() {
    String selectedCultura = (String) listaCulturas.getSelectedItemAt();
    boolean modoEstrategico = checkBoxModoEstrategico.isSelected();
    String modoJuego = radioTactico.isSelected() ? "Tctico" : "Estratgico";

    String mensaje = "Iniciando juego con Cultura: " + selectedCultura + "\nModo de
        Juego: " + modoJuego;
    if (modoEstrategico) {
        mensaje += "\nModo Estratgico Activado";
        ejecutarJuegoEstrategico();
    } else {
        mensaje += "\nModo Tctico Activado";
        JOptionPane.showMessageDialog(this, "Ejecutando lgica del juego tctico...",
            "Modo Tctico", JOptionPane.INFORMATION_MESSAGE);
        reiniciarJuego();
    }

    areaTexto.setText(mensaje);
}

private void ejecutarJuegoEstrategico() {
    // Lgica especifica del modo estrategico
    System.out.println("Ejecutando lgica del juego estrategico...");
}

private void mostrarResultado(Ejercito e1, Ejercito e2) {
    StringBuilder resultado = new StringBuilder();

    // Mostrar detalles del Ejrcito 1
    resultado.append("Ejercito 1: ").append(e1.getCultura()).append("\n");
    resultado.append("Cantidad total de Soldados:
        ").append(e1.misSoldados.size()).append("\n");
    resultado.append("Espadachines: ").append(Espadachin.cuantos()).append("\n");
    resultado.append("Arqueros: ").append(Arquero.cuantos()).append("\n");
    resultado.append("Lanceros: ").append(Lancero.cuantos()).append("\n");
    resultado.append("Caballeros: ").append(Caballero.cuantos()).append("\n\n");

    // Mostrar detalles del Ejrcito 2
    resultado.append("Ejercito 2: ").append(e2.getCultura()).append("\n");
    resultado.append("Cantidad total de Soldados:
        ").append(e2.misSoldados.size()).append("\n");
    resultado.append("Espadachines: ").append(Espadachin.cuantos()).append("\n");
    resultado.append("Arqueros: ").append(Arquero.cuantos()).append("\n");
    resultado.append("Lanceros: ").append(Lancero.cuantos()).append("\n");
    resultado.append("Caballeros: ").append(Caballero.cuantos()).append("\n\n");

    // Mostrar el nivel de poder de cada ejrcito
    resultado.append("Nivel de poder Ejercito 1: ").append(e1.poder()).append("\n");
```

```
resultado.append("Nivel de poder Ejercito 2:");
resultado.append(e2.poder()).append("\n\n");

// Mostrar el resumen del poder de ambos ejrcitos
resultado.append("El ganador es: ");
if (e1.poder() > e2.poder()) {
    resultado.append("Ejercito 1 de : ").append(e1.getCultura());
} else if (e1.poder() < e2.poder()) {
    resultado.append("Ejercito 2 de : ").append(e2.getCultura());
} else {
    resultado.append("Sin ganador");
}

areaTexto.setText(resultado.toString());
}

private Ejercito generarEjercitoAleatorio() {
    // Generar una cultura aleatoria
    String[] culturas = {"Inglaterra", "Francia", "Sacro Imperio Romano Germanico",
        "Aragon", "Moros"};
    String culturaAleatoria = culturas[new Random().nextInt(culturas.length)];

    // Generar una cantidad aleatoria de soldados (entre 1 y 10)
    int cantidadSoldados = new Random().nextInt(10) + 1;

    // Crear un ejrcito con soldados aleatorios
    return new Ejercito(culturaAleatoria, cantidadSoldados);
}

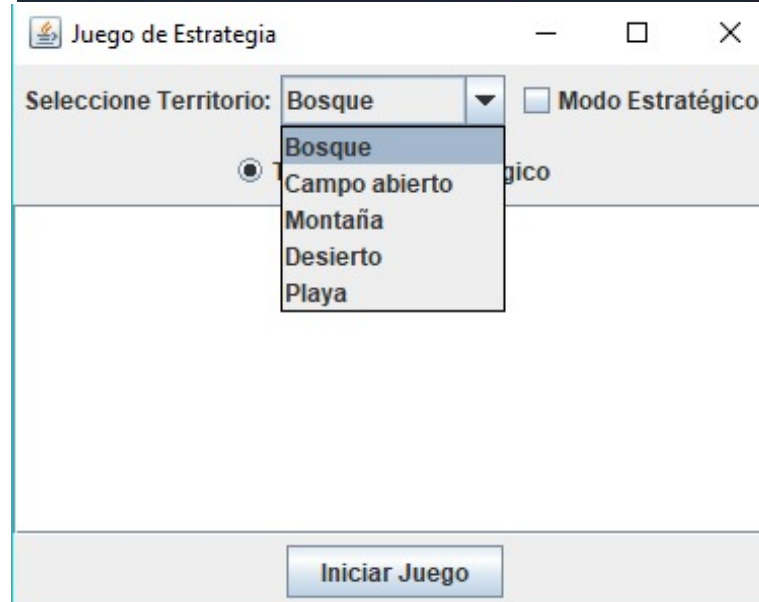
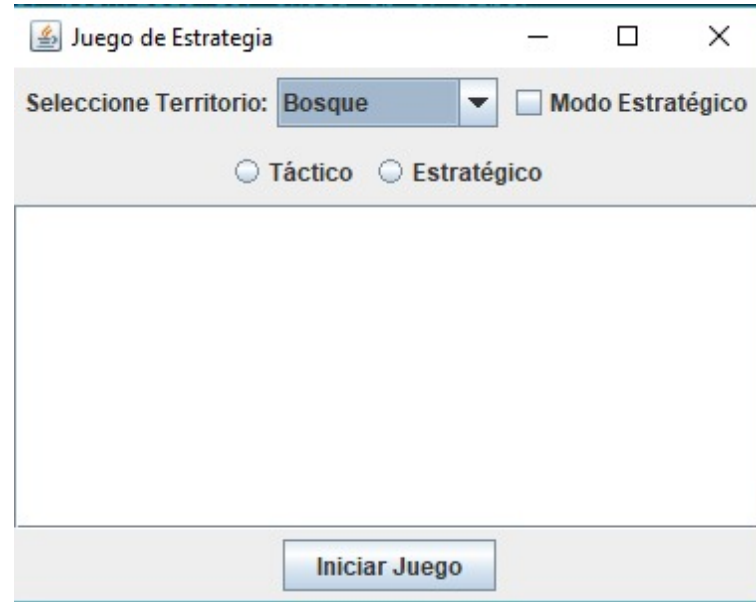
private void reiniciarJuego() {
    // Volver a ejecutar la lgica del juego y actualizar el contenido en el panel
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            panelJuego.removeAll(); // Limpiar el panel actual

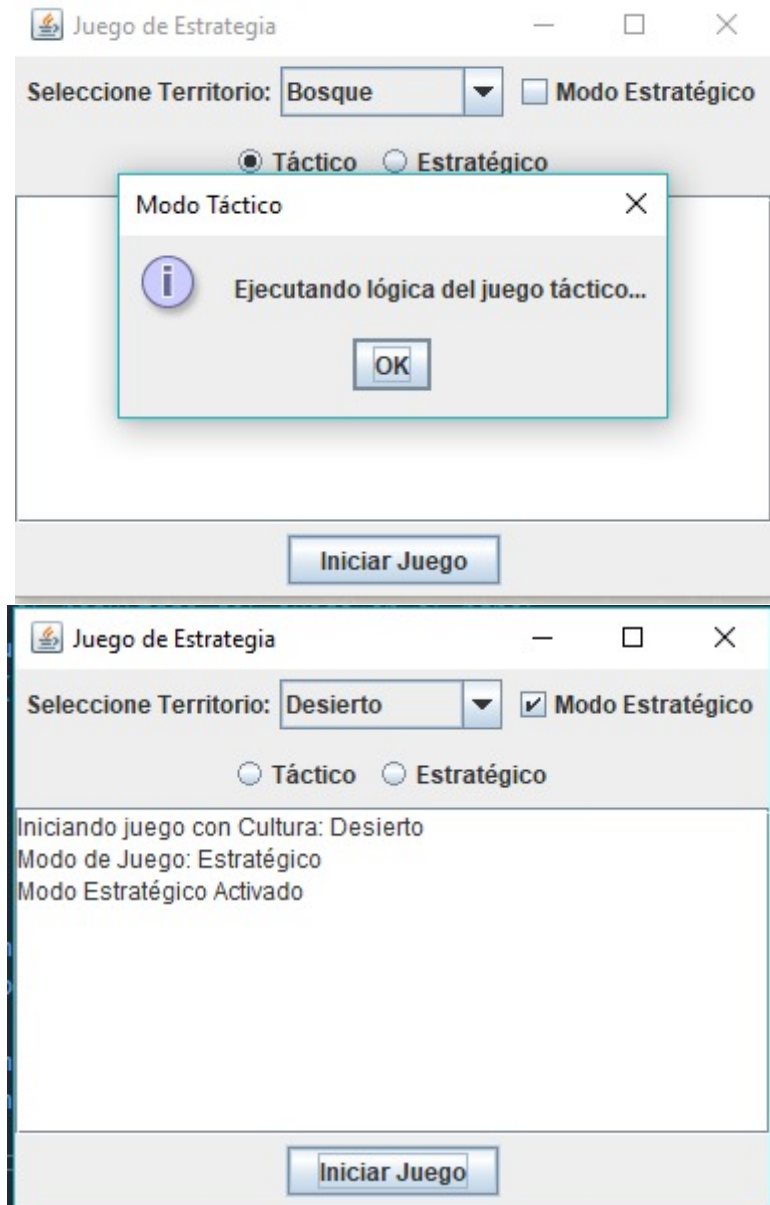
            // Generar ejrcitos aleatorios
            Ejercito e1 = generarEjercitoAleatorio();
            Ejercito e2 = generarEjercitoAleatorio();


            // Mostrar el resultado del juego en el panel
            mostrarResultado(e1, e2);
            revalidate(); // Actualizar la interfaz
            repaint();   // Repintar la interfaz
        }
    });
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new JuegoInterfazAvanzada();
        }
    });
}
```

}







Juego de Estrategia
—
□
×

Seleccione Territorio: **Desierto**
☐ Modo Estratégico

☐ Táctico
 ☒ Estratégico


Ejército 1: Sacro Imperio Romano Germanico
 Cantidad total de Soldados: 2
 Espadachines: 22
 Arqueros: 26
 Lanceros: 17
 Caballeros: 29


Juego de Estrategia
—
□
×

Seleccione Territorio: **Desierto**
☐ Modo Estratégico

☐ Táctico
 ☒ Estratégico

Ejército 2: Aragon
 Cantidad total de Soldados: 3
 Espadachines: 22
 Arqueros: 26
 Lanceros: 17
 Caballeros: 29


Juego de Estrategia
—
□
×

Seleccione Territorio: **Desierto**
☐ Modo Estratégico

☐ Táctico
 ☒ Estratégico

Cantidad total de Soldados: 3
 Espadachines: 22
 Arqueros: 26
 Lanceros: 17
 Caballeros: 29

 Nivel de poder Ejército 1: 10
 Nivel de poder Ejército 2: 23

 El ganador es: Ejército 2 de : Aragon

6. REFERENCIAS

- M. Aedo, “Fundamentos de Programación 2 - Tópicos de Programación Orientada a Objetos”, Primera Edición, 2021, Editorial UNSA.
- <https://github.com/rescobedoq/programacion.git>
- J. Dean, “Introduction to programming with Java: A Problem Solving Approach”, Third Edition, 2021, McGraw-Hill.
- C. T. Wu, “An Introduction to Object-Oriented Programming with Java”, Fifth Edition, 2010, McGraw-Hill.
- P. Deitel, “Java How to Program”, Eleventh Edition, 2017, Prentice Hall.