Informe de Laboratorio 22

Tema: Interfaz Gráfica de Usuario

Nota

Estudiante	Escuela	Asignatura
JefersonJofre Quispe Madariaga jquispemad@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Semestre: II Código: 20232200

Laboratorio	Tema	Duración
22	Interfaz Gráfica de Usuario	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 15 Enero 2024	Al 22 Enero 2024

1. Tarea

- Interfaz Gráfica de Usuario
- Usar componentes básicos GUI: Etiquetas, botones, cuadros de texto, JOptionPane, Color.
- Usar componentes avanzados GUI: Layouts, JPanel, áreas de texto, checkbox, botones de radio y combobox.

2. Equipos, materiales y temas utilizados

- Sistema Operativo Ubuntu GNU Linux 23 lunar 64 bits Kernell 6.2.
- VIM 9.0.
- OpenJDK 64-Bits 17.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.
- Programación Orientada a Objetos.
- Algoritmo de ordenamiento por inserción

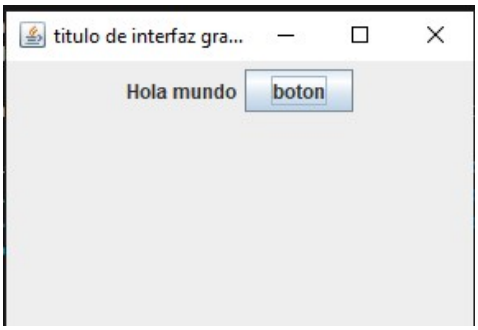
3. URL de Repositorio Github

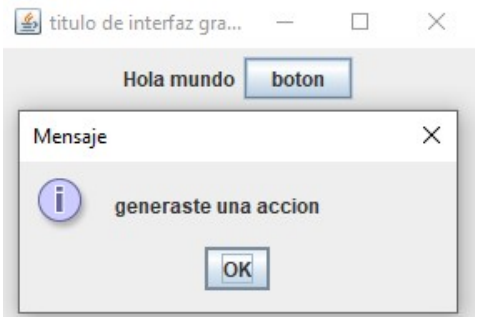
- URL del Repositorio GitHub para clonar o recuperar.
- `https://github.com/jquispemad/fp2-23b.git`
- URL para el laboratorio 22 en el Repositorio GitHub.
- `https://github.com/jquispemad/fp2-23b/tree/main/fase_03/lab_22`

4. Actividades con el repositorio GitHub

4.1. Commits de lab 22

- Mediante las importaciones con swing, awt y importandolo el awt.event.
- Importe lo anterior para crear una ventana, un panel, una etiqueta y un boton.
- Se procedio a agregar a cada objeto al panel creado.
- La primera prueba fue de crear una JFrame, un JPanel , JLabel , Jbutton con caracteristicas como el nombre de la ventana, el texto en el label y el boton con una palabra.





- A continuacion el primer codigo que use para crear una ventana agregando demas componentes .
- Primero unas importaciones generales.
- Luego instanciacion y creacion de los demas componentes.
- Se agrega los componentes al panel.

- Tambien se prioriza el tamaño.

Listing 1: codigo1.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class codigo1 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("titulo de interfaz grafica");
        JPanel panel = new JPanel();
        JLabel etiqueta = new JLabel("Hola mundo");
        JButton boton = new JButton("boton");

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        panel.add(etiqueta, BorderLayout.NORTH);
        panel.add(boton, BorderLayout.SOUTH);

        boton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(frame, "generaste una accion", "Mensaje",
                    JOptionPane.INFORMATION_MESSAGE);
            }
        });

        frame.getContentPane().add(panel);
        frame.setSize(300, 200);
        frame.setVisible(true);
    }
}
```

- Este es una parte de los codigos que use, no obstante el codigo que genera neatbeans es un poco extenso y no modificable.
- Pero me ayudo a aprender como movia los contenedores y sus objetos .
- Apartir de esto podre usar y cortar los codigos cuando los genere yo mismo
- Al principio tuve dificultades con layout pero una vez haciendo diferentes codigos y visualizarlos cada uno entendi como funcionaba.

Listing 2: Analizando codigo generado por NeatBeans

```
1  package pruebas_interfaz;
2
3
4  import javax.swing.*;
5  import java.awt.*;
6  import java.awt.event.ActionEvent;
7  import java.awt.event.ActionListener;
8
9
10
11
12 public class Menu1 extends JFrame {
13
14     private JPanel menu;
15     private JButton jugar;
16     private JButton juego_Perzonalizado;
17     private JButton salir;
18
19     public Menu1() {
20         p.setSize(800, 500);
21         initComponents();
22         this.setLocationRelativeTo(null);
23         this.setTitle("Menu");
24         this.setVisible(true);
25     }
26
27     private void initComponents() {
28
29         menu  = new JPanel();
30         jugar = new JButton();
31         juego_Perzonalizado = new JButton();
32         salir = new JButton();
33         menu.setBackground(new Color(0,0,0));
34
35         jugar.setText("Jugar");
36         jugar.addActionListener(new ActionListener() {
37             public void actionPerformed(ActionEvent evt) {
38                 Juego a = new Juego();
39                 cambiarPanel(a);
40             }
41         });
42
43         juego_Perzonalizado.setText("Juego Perzonalizado");
44         juego_Perzonalizado.addActionListener(new ActionListener() {
45             public void actionPerformed(ActionEvent evt) {
```

```
46              JuegoPersonalizado JP = new JuegoPersonalizado();
47              cambiarPanel(JP);
48          }
49      });
50
51      salir.setText("Exit");
52      salir.addActionListener(new ActionListener() {
53          public void actionPerformed(ActionEvent evt) {
54              System.exit(0);
55          }
56      });
57
58      GroupLayout MenuLayout = new GroupLayout(menu);
59      menu.setLayout(MenuLayout);
60
61      MenuLayout.setHorizontalGroup(
62          MenuLayout.createParallelGroup(GroupLayout.Alignment.LEADING)
63          .addGroup(MenuLayout.createSequentialGroup()
64              .addGap(109, 109, 109)
65              .addComponent(jugar, GroupLayout.PREFERRED_SIZE, 200,
                    GroupLayout.PREFERRED_SIZE)
66              .addGap(0, 0, 800))
67          .addGroup(GroupLayout.Alignment.TRAILING, MenuLayout.createSequentialGroup()
68              .addContainerGap(GroupLayout.DEFAULT_SIZE, 800)
69              .addComponent(salir, GroupLayout.PREFERRED_SIZE, 200,
                    GroupLayout.PREFERRED_SIZE)
70              .addGap(71, 71, 71))
71          .addGroup(MenuLayout.createSequentialGroup()
72              .addGap(298, 298, 298)
73              .addComponent(juego_Perzonalizado, GroupLayout.PREFERRED_SIZE, 200,
                    GroupLayout.PREFERRED_SIZE)
74              .addContainerGap(302, 800))
75      );
76      MenuLayout.setVerticalGroup(
77          MenuLayout.createParallelGroup(GroupLayout.Alignment.LEADING)
78          .addGroup(MenuLayout.createSequentialGroup()
79              .addGap(132, 132, 132)
80              .addComponent(jugar, GroupLayout.PREFERRED_SIZE, 45,
                    GroupLayout.PREFERRED_SIZE)
81              .addGap(60, 60, 60)
82              .addComponent(juego_Perzonalizado, GroupLayout.PREFERRED_SIZE, 45,
                    GroupLayout.PREFERRED_SIZE)
83              .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED, 100, 800)
84              .addComponent(salir, GroupLayout.PREFERRED_SIZE, 45,
                    GroupLayout.PREFERRED_SIZE)
85              .addGap(73, 73, 73))
86      );
87
88      GroupLayout layout = new GroupLayout(getContentPane());
89      getContentPane().setLayout(layout);
90      layout.setHorizontalGroup(
91          layout.createParallelGroup(GroupLayout.Alignment.LEADING)
92          .addComponent(menu, GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, 800)
93      );
94      layout.setVerticalGroup(
95          layout.createParallelGroup(GroupLayout.Alignment.LEADING)
```

```
            .addComponent(menu, GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, 800)
        );

        menu.getAccessibleContext().setAccessibleName("");

        pack();
    }


    private void cambiarPanel(JPanel p){
        p.setSize(800, 500);
        p.setLocation(0,0);
        menu.removeAll();
        menu.add(p);
        menu.revalidate();
        menu.repaint();
    }
}
```

- En este fragmento de codigo Java,he creado una interfaz grafica de un tablero utilizando Java Swing y AWT. La interfaz incluye una cuadricula de 100 botones dispuestos en un panel (jPanel1) y botones de direccion (arriba, abajo, izquierda, derecha) en otro panel (jPanel2).
- La creacion que me reduce el codigo es con un array. Cada boton se identifica con un nombre unico utilizando la formula setName((i
- Se implementaron manejadores de eventos para los botones de direccion, registrando la direccion seleccionada (arriba, abajo, izquierda, derecha) al ser clickeados.
- Al hacer clic en cualquier boton en el area principal (jPanel1), se ejecuta el metodo botonActionPerformed, actualizando las coordenadas x e y del juego segun la posicion del boton clickeado.
- Las ubicaciones de los componentes en la interfaz se logra mediante el uso de GroupLayout, simplificando la estructura y organizacion del codigo.
- Demostrando que puedo crear y ubicar objetos en un interfaz grafica

Listing 3: Creando la interfaz de tablero

```

package pruebas_interfaz;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Juego extends javax.swing.JPanel {
    private JButton[] boton = new JButton[100];
    private JButton jugar = new JButton();
    private JButton arriba = new JButton();
    private JButton abajo = new JButton();
    private JButton derecha = new JButton();
    private JButton izquierda = new JButton();
    private int x, y, direccion;
    private JPanel jPanel1;
    private JPanel jPanel2;
```

```
133
134     public Juego() {
135         initComponents();
136     }
137     private void initComponents() {
138 
139         jPanel1 = new JPanel(new GridLayout(10, 10));
140         boton = new JButton[100];
141         for (int i = 0; i < 100; i++) {
142             boton[i] = new JButton();
143             boton[i].setName((i%10) + "" + ((99-i)/10));
144             jPanel1.add(boton[i]);
145             boton[i].setBorder(BorderFactory.createEtchedBorder());
146             boton[i].addActionListener(new ActionListener() {
147                 public void actionPerformed(ActionEvent e) {
148                     botonActionPerformed(e);
149                 }
150 
151             });
152         }
153         jPanel2 = new JPanel(new GridLayout(2,2));
154         jugar = new JButton();
155         arriba = new JButton();
156         abajo = new JButton();
157         derecha = new JButton();
158         izquierda = new JButton();
159 
160         arriba.setText("ar");
161         arriba.addActionListener(new ActionListener() {
162             public void actionPerformed(ActionEvent evt) {
163                 direccion = 2;
164                 System.out.println(2);
165             }
166         });
167 
168         izquierda.setText("iz");
169         izquierda.addActionListener(new ActionListener() {
170             public void actionPerformed(ActionEvent evt) {
171                 direccion = 1;
172                 System.out.println(1);
173             }
174         });
175 
176         derecha.setText("de");
177         derecha.addActionListener(new ActionListener() {
178             public void actionPerformed(ActionEvent evt) {
179                 direccion = 4;
180                 System.out.println(4);
181             }
182         });
183 
184         abajo.setText("ab");
185         abajo.addActionListener(new ActionListener() {
186             public void actionPerformed(ActionEvent evt) {
187                 direccion = 3;
188                 System.out.println(3);
```

```
189             }
190         });
191 
192 
193         GroupLayout layout = new GroupLayout(this);
194         this.setLayout(layout);
195         layout.setHorizontalGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
196                 .addGroup(layout.createSequentialGroup()
197                 .addComponent(jPanel1, GroupLayout.PREFERRED_SIZE, 500,
                        GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 77,
                        Short.MAX_VALUE)
198                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
199                     .addGroup(layout.createSequentialGroup()
200                         .addComponent(izquierda, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
                                javax.swing.GroupLayout.PREFERRED_SIZE)
201                         .addGap(56, 56, 56))
202                     .addComponent(arriba, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
                            javax.swing.GroupLayout.PREFERRED_SIZE)
203                     .addComponent(abajo, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
                            javax.swing.GroupLayout.PREFERRED_SIZE))
204                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
205                 .addComponent(derecha, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
206                 .addGap(61, 61, 61)));
207         layout.setVerticalGroup( layout.createParallelGroup(GroupLayout.Alignment.LEADING)
208                 .addComponent(jPanel1, GroupLayout.DEFAULT_SIZE, 500, Short.MAX_VALUE)
209                 .addGroup(layout.createSequentialGroup()
210                 .addGap(132, 132, 132)
211                 .addComponent(arriba, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
212                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
213                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
214                     .addComponent(derecha, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
                            javax.swing.GroupLayout.PREFERRED_SIZE)
215                     .addComponent(izquierda, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
                            javax.swing.GroupLayout.PREFERRED_SIZE))
216                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
217                 .addComponent(abajo, javax.swing.GroupLayout.PREFERRED_SIZE, 50,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
218                 .addGap(0, 206, Short.MAX_VALUE)));
219     }
220 
221     private void botonActionPerformed(ActionEvent e) {
222         int num = Integer.parseInt(((JButton)e.getSource()).getName());
223         this.x = (num/10) +1;
224         this.y = (num%10) +1;
225     }
226 
227 
228     public void casilla(){
229 
230     }
231 }
```

- Hasta este punto estuve modificando las variables y reordenandolo para que pueda ser un poco

mas facil de comprender aunque aun me falta.

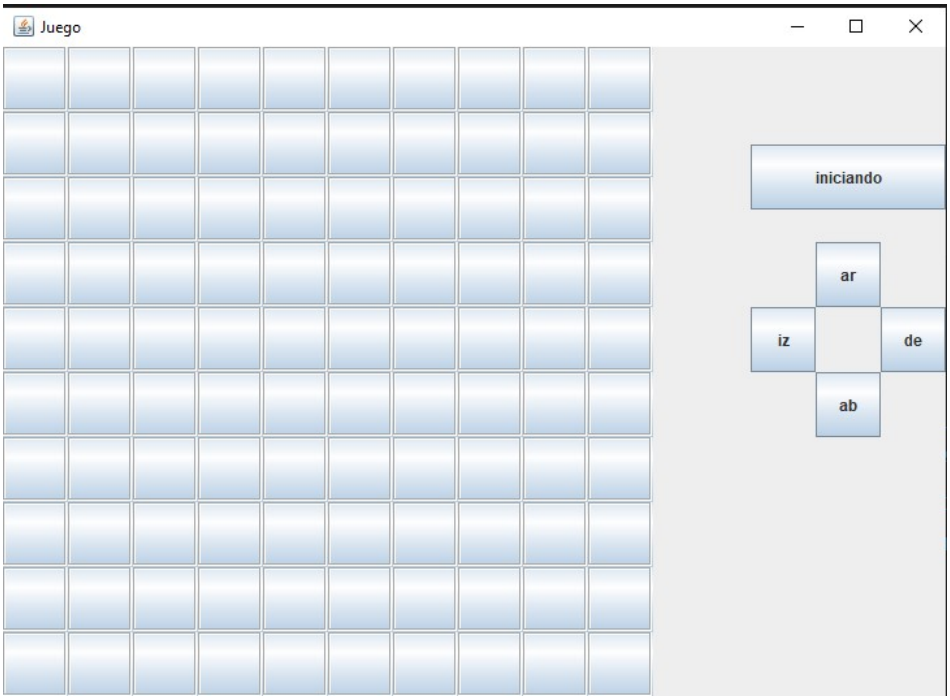
- Primero en posicion de elementos:
- Utilice principalmente 3 cosas para agregar juntar y posicionar
- Primero para agregar cosas es ".addComponentïngresando 4 el componente y el tamaño
- Segundo para agrupar los componentes de forma paralela ".addGroup(layout.createSequentialGroup()"
- Y ultimo para poder posicionar es ".addGapïngresando 3 numeros min, "distancia adecuadaz la distancia era la que yo ingresaba
- Todo esto con 2 metodos .setHorizontal(layout.createParallelGroup(... y .setVertical(layout.createParallelGroup(... que era para ubicarlo de manera horizontal y vertical
- .addGap lo posiciona desde arriba a la izquierda hasta abajo a la derecha en esas direcciones

Listing 4: codigo1.java

```
1
2  import javax.swing.*;
3  import java.awt.*;
4  import java.awt.event.ActionEvent;
5  import java.awt.event.ActionListener;
6
7  public class codigo2 extends JPanel {
8      private JButton[] boton = new JButton[100];
9      private JButton jugar = new JButton();
10     private JButton arriba = new JButton();
11     private JButton abajo = new JButton();
12     private JButton derecha = new JButton();
13     private JButton izquierda = new JButton();
14     private int x, y, direccion;
15     private JPanel jPanel1;
16     private JButton iniciarTurno = new JButton();
17
18     public codigo2() {
19         initComponents();
20     }
21     private void initComponents() {
22
23         jPanel1 = new JPanel(new GridLayout(10, 10));
24         int lado = 500;
25         boton = new JButton[100];
26         for (int i = 0; i < 100; i++) {
27             boton[i] = new JButton();
28             boton[i].setName((i%10) + "" + ((99-i)/10));
29             jPanel1.add(boton[i]);
30             boton[i].setBorder(BorderFactory.createEtchedBorder());
31             boton[i].addActionListener(new ActionListener() {
32                 public void actionPerformed(ActionEvent e) {
33                     botonActionPerformed(e);
34                 }
35
36             });
37         }
```

```
38          int ladoBoton = 50;
39          iniciarTurno = new JButton();
40          jugar = new JButton();
41          arriba = new JButton();
42          abajo = new JButton();
43          derecha = new JButton();
44          izquierda = new JButton();
45          arriba.setText("ar");
46          arriba.addActionListener(new ActionListener() {
47              public void actionPerformed(ActionEvent evt) {
48                  direccion = 2;
49                  System.out.println(2);
50              }
51          });
52          izquierda.setText("iz");
53          izquierda.addActionListener(new ActionListener() {
54              public void actionPerformed(ActionEvent evt) {
55                  direccion = 1;
56                  System.out.println(1);
57              }
58          });
59          derecha.setText("de");
60          derecha.addActionListener(new ActionListener() {
61              public void actionPerformed(ActionEvent evt) {
62                  direccion = 4;
63                  System.out.println(4);
64              }
65          });
66          abajo.setText("ab");
67          abajo.addActionListener(new ActionListener() {
68              public void actionPerformed(ActionEvent evt) {
69                  direccion = 3;
70                  System.out.println(3);
71              }
72          });
73          iniciarTurno.setText("iniciando");
74          iniciarTurno.addActionListener(new ActionListener() {
75              public void actionPerformed(ActionEvent evt) {
76                  //funcion movimietno en el reino
77                  //funcion actualizacion de visualizacion del tablero segun la lista del reino
78                  System.out.println("Comenzando");
79                  repaint();
80              }
81          });
82
83          GroupLayout layout = new GroupLayout(this);
84          this.setLayout(layout);
85          layout.setHorizontalGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
86                  .addComponent(jPanel1, lado, lado, lado)
87                  .addGroup(layout.createSequentialGroup()
88                      .addGap(lado+75,lado+75,lado+75)
89                      .addComponent(izquierda, ladoBoton, ladoBoton, ladoBoton)
90                      .addGroup(layout.createParallelGroup(GroupLayout.Alignment.TRAILING)
91                          .addComponent(arriba, ladoBoton, ladoBoton, ladoBoton)
92                          .addComponent(abajo , ladoBoton, ladoBoton, ladoBoton))
93                      .addComponent(derecha, ladoBoton, ladoBoton, ladoBoton))
```

```
94                  .addGroup(layout.createSequentialGroup()
95                      .addGap(lado, lado +75, lado+ 75)
96                      .addComponent(iniciarTurno, ladoBoton, ladoBoton + 100, ladoBoton+100)
97                  )
98          );
99
100         layout.setVerticalGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
101                 .addComponent(jPanel1, lado, lado, lado)
102                 .addGroup(layout.createSequentialGroup()
103                     .addGap(150, 150, 150)
104                     .addComponent(arriba, ladoBoton, ladoBoton, ladoBoton)
105                     .addGroup(layout.createParallelGroup(GroupLayout.Alignment.TRAILING)
106                         .addComponent(derecha, ladoBoton, ladoBoton, ladoBoton)
107                         .addComponent(izquierda, ladoBoton, ladoBoton, ladoBoton))
108                     .addComponent(abajo, ladoBoton, ladoBoton, ladoBoton))
109                 .addGroup(layout.createSequentialGroup()
110                     .addGap(75,75,75)
111                     .addComponent(iniciarTurno, ladoBoton, ladoBoton, ladoBoton)
112                 )
113         );
114     }
115
116     private void botonActionPerformed(ActionEvent e) {
117         int num = Integer.parseInt(((JButton)e.getSource()).getName());
118         this.x = (num/10) +1;
119         this.y = (num%10) +1;
120     }
121     public void casilla(){
122
123     }
124 }
```



- En el siguiente codigo se puede ver que se agrego colores a los botones ademas de funcionalidades para empezar el juego
- El uso de setBackground(color ..) para los botones o para jpanel.
- Uso de action listener para conectar los botones con el videojuego
- Se coloco solo un boton para re pintar los botones y para actualizar la lista del ejercito con sus posiciones

Listing 5: Creando la interfaz de tablero

```
232
233 import javax.swing.*;
234 import java.awt.*;
235 import java.awt.event.ActionEvent;
236 import java.awt.event.ActionListener;
237 import java.lang.reflect.Array;
238 import java.util.ArrayList;
239
240 public class Juego extends JPanel {
241     private JButton[] boton = new JButton[100];
242     private boolean turno;
243     private JButton arriba;
244     private JButton abajo;
245     private JButton derecha;
246     private JButton izquierda;
247     private int x, y, direccion, x_x, y_y;
248     private JPanel jPanel1;
249     private JButton iniciarTurno;
250     private Reinos reinos;
251
```

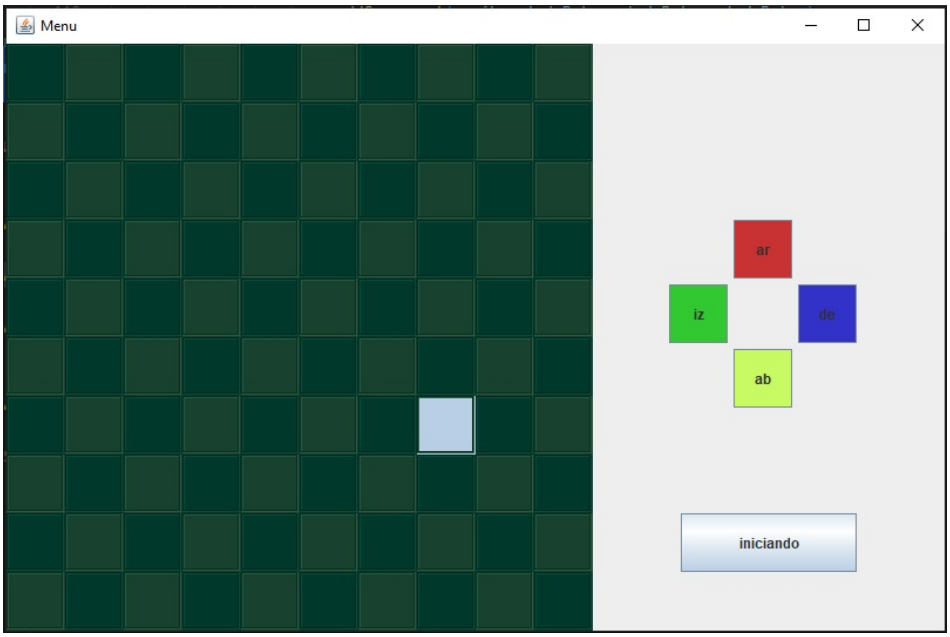
```
252     public Juego() {
253         initComponents();
254         reinos = new Reinos();
255         reinos.Reinos();
256         this.turno = true;
257         System.out.println(reinos.toString());
258     }
259 
260 
261     private void initComponents() {
262         Color colora = new Color(0, 56, 43);
263         Color colorb = new Color(25, 65, 47);
264         jPanel1 = new JPanel(new GridLayout(10, 10));
265         int lado = 500;
266         boton = new JButton[100];
267         for (int i = 0; i < 100; i++) {
268             boton[i] = new JButton();
269             if((i%2==0 && i/10%2 == 0) || (i/10%2==1 && i%2==1 )){
270                 boton[i].setBackground(colora);
271             }else{
272                 boton[i].setBackground(colorb);
273             }
274             boton[i].setName((i%10) + "" + ((99-i)/10));
275             jPanel1.add(boton[i]);
276             boton[i].setBorder(BorderFactory.createEtchedBorder());
277             boton[i].addActionListener(new ActionListener() {
278                 public void actionPerformed(ActionEvent e) {
279                     botonActionPerformed(e);
280                 }
281 
282             });
283         }
284         int ladoBoton = 50;
285         iniciarTurno = new JButton();
286         arriba = new JButton();
287         abajo = new JButton();
288         derecha = new JButton();
289         izquierda = new JButton();
290 
291         arriba.setBackground(new Color(200,50,50));
292         arriba.setText("ar");
293         arriba.addActionListener(new ActionListener() {
294             public void actionPerformed(ActionEvent evt) {
295                 direccion = 2;
296                 System.out.println(2);
297             }
298         });
299 
300         izquierda.setBackground(new Color(50,200,50));
301         izquierda.setText("iz");
302         izquierda.addActionListener(new ActionListener() {
303             public void actionPerformed(ActionEvent evt) {
304                 direccion = 1;
305                 System.out.println(1);
306             }
307         });
```

```
308
309         derecha.setBackground(new Color(50,50,200));
310         derecha.setText("de");
311         derecha.addActionListener(new ActionListener() {
312             public void actionPerformed(ActionEvent evt) {
313                 direccion = 4;
314                 System.out.println(4);
315             }
316         });
317 
318         abajo.setBackground(new Color(200,250,100));
319         abajo.setText("ab");
320         abajo.addActionListener(new ActionListener() {
321             public void actionPerformed(ActionEvent evt) {
322                 direccion = 3;
323                 System.out.println(3);
324             }
325         });
326         iniciarTurno.setText("iniciando");
327         iniciarTurno.addActionListener(new ActionListener() {
328             public void actionPerformed(ActionEvent evt) {
329                 moviendoReino(evt);
330                 repaint();
331             }
332         });
333 
334         GroupLayout layout = new GroupLayout(this);
335         this.setLayout(layout);
336         layout.setHorizontalGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
337                 .addComponent(jPanel1, lado, lado, lado)
338                 .addGroup(layout.createSequentialGroup()
339                     .addGap(lado+65,lado+65,lado+65)
340                     .addComponent(izquierda, ladoBoton, ladoBoton, ladoBoton)
341                     .addGap(5, 5, 5)
342                     .addGroup(layout.createParallelGroup(GroupLayout.Alignment.TRAILING)
343                         .addComponent(arriba, ladoBoton, ladoBoton, ladoBoton)
344                         .addComponent(abajo , ladoBoton, ladoBoton, ladoBoton))
345                     .addGap(5, 5, 5)
346                     .addComponent(derecha, ladoBoton, ladoBoton, ladoBoton))
347                 .addGroup(layout.createSequentialGroup()
348                     .addGap(lado, lado +75, lado+ 75)
349                     .addComponent(iniciarTurno, ladoBoton, ladoBoton + 100, ladoBoton+100)
350                 )
351         );
352         layout.setVerticalGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
353                 .addComponent(jPanel1, lado, lado, lado)
354                 .addGroup(layout.createSequentialGroup()
355                     .addGap(150, 150, 150)
356                     .addComponent(arriba, ladoBoton, ladoBoton, ladoBoton)
357                     .addGap(5, 5, 5)
358                     .addGroup(layout.createParallelGroup(GroupLayout.Alignment.TRAILING)
359                         .addComponent(derecha, ladoBoton, ladoBoton, ladoBoton)
360                         .addComponent(izquierda, ladoBoton, ladoBoton, ladoBoton))
361                     .addGap(5, 5, 5)
362                     .addComponent(abajo, ladoBoton, ladoBoton, ladoBoton))
363                 .addGroup(layout.createSequentialGroup()
```

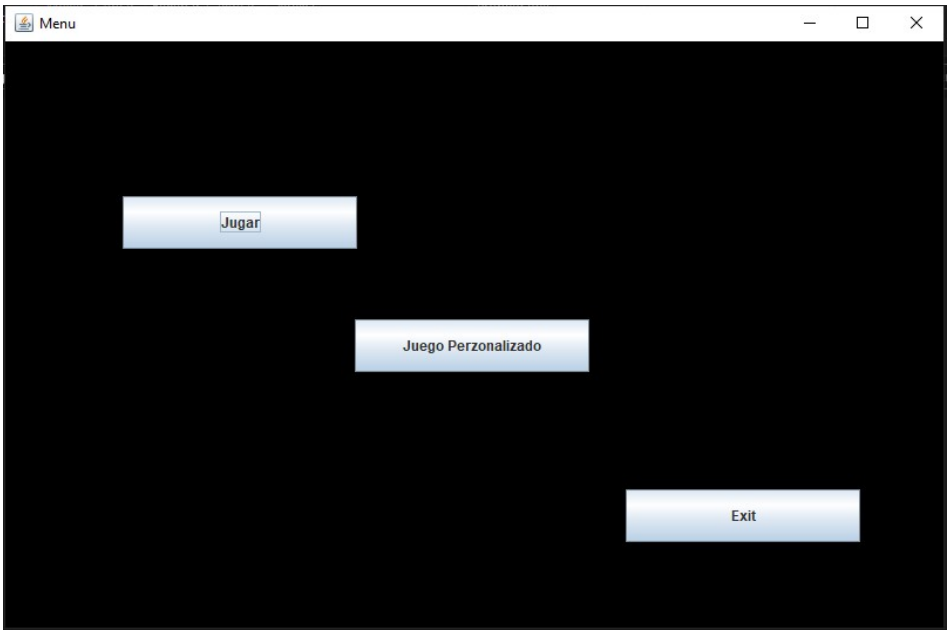
```
364                 .addGap(400,400,400)
365                 .addComponent(iniciarTurno, ladoBoton, ladoBoton, ladoBoton)
366             )
367     );
368 }
369 
370 
371 //acciones y funciones de los botones
372 private void botonActionPerformed(ActionEvent e) {
373     int num = Integer.parseInt(((JButton)e.getSource()).getName());
374     this.x = (num/10) +1;
375     this.y = (num%10) +1;
376 }
377 private void moviendoReino(ActionEvent e) {
378     if(reinos.verificarPosicionLibreEjercitos(x , y)){//veridficar si la casilla que
          presiono tiene un ejercito
379         System.out.println("-----" + reinos.verificarPosicionLibreEjercitos(x, y));
380         x_x = x;
381         y_y = y;
382         movimientoCopia();
383         if(reinos.verificarPosicionLibreEjercitos(x_x, y_y)){
384             System.out.println("estuvo libre la siguiente posicion");
385             movimiento();
386         }else if(reinos.verificarAmigoPosicion(x_x, y_y,
              reinos.getReinos().get(reinos.buscarEjercito(x_x,y_y)))){
387             System.out.println("Hay un amigo en la siguiente posicion");
388         }else{
389             System.out.println("Enemigo a la vista");
390             movimiento();
391         }
392         System.out.println(reinos.toString());
393 
394         if(true)
395             turno = false;
396 
397         if(reinos.verificarReinos())
398             System.exit(0);
399     }
400 }
401 private void movimiento(){
402     switch (direccion) {
403         case 1: this.x--; break;//izquierda
404         case 2: this.y++; break;//arriba
405         case 3: this.y--; break;//abajo
406         case 4: this.x++; break;//derecha
407         default:System.out.println("Error: movimiento() --> Juego"); break;
408     }
409 }
410 private void movimientoCopia(){
411     switch (direccion) {
412         case 1: x_x--; break;//izquierda
413         case 2: y_y++; break;//arriba
414         case 3: y_y--; break;//abajo
415         case 4: x_x++; break;//derecha
416         default:System.out.println("Error: movimiento() --> Juego-otro"); break;
417     }
```

```
418     }
419 }
```

- Ejecucion del main y visualizacion de ventana de juego



- Desde el menu direccionar al panel del juego



4.2. Estructura de laboratorio 22

- El contenido que se entrega en este laboratorio es el siguiente:

```
lab_22
|---Arquero.java
|---Caballero.java
|---Ejercito.java
|---Espadachin.java
|---Juego.java
|---Lancero.java
|---Latex
| |---Nueva carpeta
| |---programacion_lab22__jquispemad_v1.0.log
| |---programacion_lab22__jquispemad_v1.0.pdf
| |---programacion_lab22__jquispemad_v1.0.synctex.gz
| |---programacion_lab22__jquispemad_v1.0.tex
| |---src
| | |---codigo1.java
| | |---img_primer_dos.jpg
| | |---img_primer_uno.jpg
| | |---insercion.png
| | |---logo_abet.png
| | |---logo_episunsa.png
| | |---logo_unsa.jpg
| | |---ventana_de_juego.jpg
| | |---ventana_menu.jpg
|---Main.java
|---Mapa.java
|---Menu.java
|---pruebas_interfaz
| |---JuegoPersonalizado.java
| |---PanelFondo.java
| |---pruebas.java
|---Reinos.java
|---Soldado.java
|---Tablero.java
```

5. Rúbricas

5.1. Entregable Informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y facil de leer.

5.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumplio 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 items.

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

Tabla 2: 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

Tabla 3: Rúbrica para contenido del Informe y demostración

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	4	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1.5	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
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 estan dentro de los plazos de fecha de entrega establecidos.	2	X	2	
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	2	
Total		20		17.5	

6. Referencias

- https://www.w3schools.com/java/default.asp
- https://www.geeksforgeeks.org/insertion-sort/