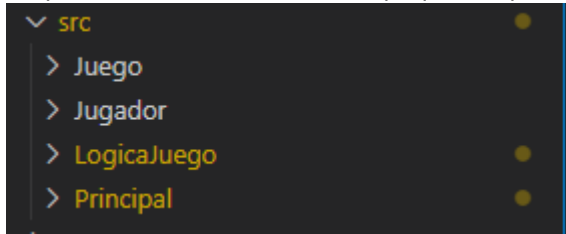


Manual técnico

Empezaremos describiendo los paquetes que encontraremos dentro del código.



Encontraremos cuatro paquetes,

El paquete juego será el encargado de desarrollar las interfaces graficas para los menús que se cuentan dentro del juego.

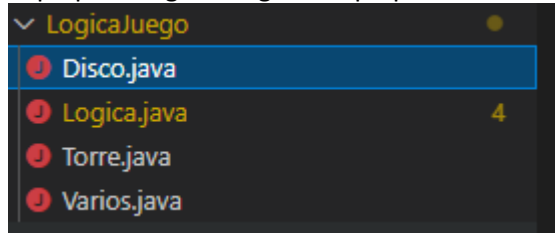


Encontraremos las diferentes clases para poder crear los paneles gráficos así mismo como la clase que nos ayuda a crear la ventana donde estos mismos serán agregados.

El paquete jugador es el que nos ayudara a la creación de objetos tipo usuario, para así poder llenar las estadísticas.

```
public Usuario(String nombre, int victorias, int derrotas, int abandonadas, int totalMovimientos, int noPartidas, int tiempoPromedio) {
    this.nombre = nombre;
    this.victorias = victorias;
    this.derrotas = derrotas;
    this.abandonadas = abandonadas;
    this.totalMovimientos = totalMovimientos;
    this.noPartidas = noPartidas;
    this.tiempoPromedio = tiempoPromedio;
}
```

El paquete lógica Juego es el paquete donde todo el desarrollo del juego se lleva a cabo



En la clase disco encontraremos únicamente la función de crear los discos y agregarles un nuevo color aleatoriamente

```
public Disco() {
    Random color = new Random();

    float color1 = color.nextFloat();
    float color2 = color.nextFloat();
    float color3 = color.nextFloat();

    Color colorDisco = new Color(color1, color2, color3);
    this.setBackground(colorDisco);
}
```

En la clase lógica encontraremos básicamente toda la lógica del juego, encontraremos opciones como la de agregar los discos

```
public void AgregarDiscos(int n) {
    Disco disco = new Disco();
    disco.setBounds(x: 90, y: 250, width: 45, height: 20);
    disco.addMouseListener((MouseListener) this);
    torre1.add(disco);
    torre1.updateUI();

    for (int i = 1; i <= n - 1; i++) {
        torre1.add(new Disco());
        torre1.getComponent(i).addMouseListener(this);
    }
    Ordenar(n);
    torre1.updateUI();
}
```

De ordenar estos mismos con otro método.

```

public void Ordenar(int n) {
    if (n >= 0) {
        for (int j = 1; j <= n - 1; j++) {

            JPanel discoAnterior = (JPanel) torre1.getComponent(j - 1);
            int x = discoAnterior.getX();
            int y = discoAnterior.getY();
            int w = discoAnterior.getWidth();
            int h = discoAnterior.getHeight();

            JPanel disco = (JPanel) torre1.getComponent(j);
            disco.setBounds(x, y - h, w, h);
            discoAnterior.setBounds(x - 10, y, w + 20, h);
            torre1.setComponentZOrder(disco, j);
            torre1.setComponentZOrder(discoAnterior, j - 1);

        }
        Ordenar(n - 1);
    }
}

```

Y también encontraremos el desarrollo de todo el código para agregar las reglas básicas al juego. En la clase torre al igual que la clase disco solo se encarga de darle forma a estos objetos

```

public Torre() {
    this.setLayout(mgr: null);
    this.setBackground(Color.PINK);
}

@Override
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setColor(Color.BLACK);
    g.fillRect(x: 10, y: 270, width: 200, height: 15);
    g.fillRect(x: 105, y: 30, width: 15, height: 240);
}
}

```

En este caso dibujamos las torres sobre un panel que luego llamamos a la clase lógica y así sucesivamente.

Y la ultima clase de este paquete es la clase varios en ella encontraremos algunas opciones como la de verificar si ya se termino el juego o la de ir contando los pasos

```

public boolean verificarFinalJuego(int n, int numeroArosTorre3) {
    return n == numeroArosTorre3;
}

public int getContador() {
    return contador;
}

public void setContador(int contador) {
    this.contador = contador;
}
}

```

Y por ultimo tendremos la clase principal en ella encontraremos la única función de mandar a llamar la parte grafica del juego para así poder ejecutarla A

```

public static void main(String[] args){
    System.out.println(x: "Principal.Principal.main()");
    Ventana play = new Ventana();
}
}

```