FLAPPY BIRD
MANUAL DEL PROGRAMADOR

Desarrolladores:

Espinoza M., Julián C.
Herrera V., Santiago
Lozada M., Jesús A.

Tabla de contenido

Tabla de códigos fuente

Introducción

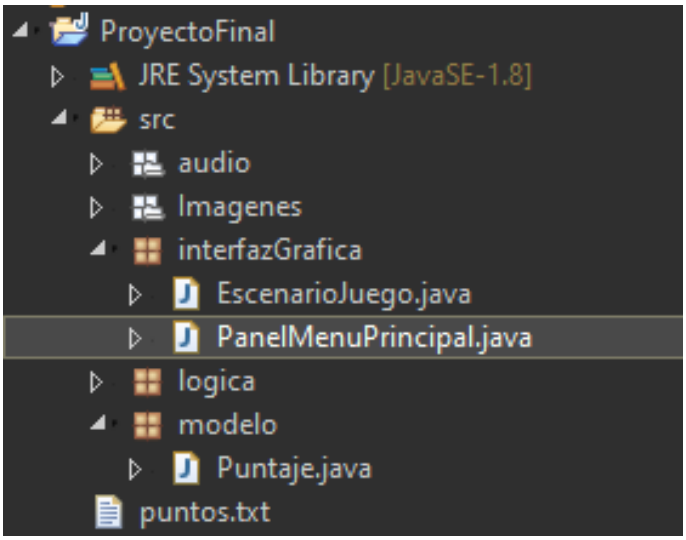
El propósito de este manual del programador, es dar a conocer al lector los códigos y clases utilizadas en el desarrollo del juego Flappy Bird. Para ello se busca de la forma más concisa, explicar cada uno de los códigos junto con la logica utilizada en el desarrollo del software.

1. Estructura

Este proyecto se divide estructuralmente en 3 partes importantes o dicho de otro modo, se encuentra diseñado con una estructura de 3 capas, en la que se dividen las clases en 3 paquetes principales, de la siguiente manera:

- ❖ logica (Lógica): En ella están contenidas todas las clases que tienen un componente de logica, procesamiento de información o que realice algún cambio en el comportamiento del programa.
- ❖ modelo (Persistencia): Contiene la clase encargada de la persistencia de datos, como la creación, actualización y recuperación de la información.
- ❖ interfazGrafica (GUI): Incluye todos los componentes gráficos, con los que interactúa el usuario, como los métodos necesarios para la correcta visualización de los elementos y su actualización en pantalla.

En este manual se tratara de explicar lo mejor posible las clases que conforman estos paquetes anteriormente mencionados.



2. Librerías

En total las librerías utilizadas para el desarrollo de este juego se pueden resumir en 5 principales, las cuales son:

- javax.swing.*: Utilizada para la creación de los múltiples elementos gráficos que componente la interfaz.
- java.awt.*: Esta librería permite acceder a varias funcionalidades para el manejo de textos e imágenes, así como de listeners utilizados.
- java.io.*: es la librería que permite hacer la conexión con archivos externos permitiendo así la persistencia de los datos.
- java.util.ArrayList: Librería que permite la utilización de los ArrayList.
- javax.sound.sampled.*: Utilizada para implementar todos los sonidos del proyecto.

3. Paquete modelo

Este paquete almacena la clase encargada de realizar la persistencia de los datos del juego, contiene una única clase encargada de la tarea anteriormente mencionada.

3.1. Puntaje.java: es la que logra la persistencia de los datos.

```
package modelo;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.ArrayList;

/**
 * Declaración de clase Puntaje
 *
 * @author Santiago Herrera
 * @version
 *
 */
public class Puntaje implements Serializable {
	private static final long serialVersionUID = 1L;

	// ArrayLists
	private ArrayList<Integer> puntajes = new ArrayList<Integer>();

	// Constructs
	/**
	 * Constructor Puntaje
	 */
	@SuppressWarnings("unchecked")
	public Puntaje() {
		try {
			ObjectInputStream ois = new ObjectInputStream(new
			FileInputStream("puntos.txt"));
```

```
                    Object obj = ois.readObject();

                    if (obj != null) {
                        if (obj instanceof ArrayList<?>) {
                            if (((ArrayList<?>) obj).get(0) instanceof Integer) {
                                this.puntajes = (ArrayList<Integer>) obj;
                                this.ordenarPuntos();
                            }
                        }
                    }
                    ois.close();
                } catch (FileNotFoundException fnfe) {
                } catch (IOException ioe) {
                } catch (ClassNotFoundException cnfe) {
                }
        }

        // Methods
        /**
         *
         * @param puntos
         */
        public void addPuntos(int puntos) {
                File file = new File("puntos.txt");
                puntajes.add(puntos);

                if (file.exists()) {
                        file.delete();
                }
                try {
                        ObjectOutputStream oos = new ObjectOutputStream(new
                        FileOutputStream("puntos.txt"));
                        oos.writeObject(this.puntajes);
                        oos.close();
                } catch (FileNotFoundException fnfe) {
                } catch (IOException ioe) {
                }
        }

        /**
         *
         */
        public void ordenarPuntos() {
                int i;
                int j;
                int aux;
```

```
                for (i = 0; i < this.puntajes.size(); i++) {
                        for (j = 0; j < this.puntajes.size(); j++) {
                                if (this.puntajes.get(i) > this.puntajes.get(j)) {
                                        aux = this.puntajes.get(i);
                                        this.puntajes.set(i, this.puntajes.get(j));
                                        this.puntajes.set(j, aux);
                                }
                        }
                }
        }

        // Getters
        public ArrayList<Integer> getPuntajes() {
                return puntajes;
        }
}
```

Código 1 Clase Puntaje.java

4. Paquete lógica

La logica del programa está formada por todas las clases que sirven para el funcionamiento interno del juego, como el manejo de los hilos, el movimiento de los distintos componentes de la pantalla así como el paso de información entre clases que permitan realizar los cálculos y la toma de decisiones pertinentes durante la ejecución del programa.

4.1. MainClass.java: Clase principal del programa que llama e instancia los objetos necesarios para la ejecución del programa.

```
package logica;

import interfazGrafica.EscenarioJuego;
import modelo.Puntaje;

/**
 * Declaracion clase MainClass
 *
 * @author Julian Espinoza
 *
 */
public class MainClass {
	/**
	 * Metodo main
	 *
	 * @param args
	 */
	public static void main(String[] args) {
		Puntaje db = new Puntaje();

		@SuppressWarnings("unused")
		EscenarioJuego juego = new EscenarioJuego(db);
	}
}
```

Código 2 Clase MainClass.java

4.2. Hilo.java: Clase abstracta que indica los métodos que deberá ser obligatorio sobrescribir en las subclases que hereden de ella.

```
package logica;

/**
 * Declaracion clase Hilo
 *
 * @author Santiago Herrera.
 *
 */
public abstract class Hilo extends Thread {

	public abstract void resumeThread();

	public abstract void pauseThread();
}
```

Código 3 Clase Hilo.java

4.3. Pajaro.java: Clase donde se guarda toda la logica relacionada con el pájaro y que a su vez hereda de Hilo, se encarga de controlar el movimiento del pájaro y de la animación asociada a él, define los límites del movimiento y los sonidos que realiza el pájaro al momento de saltar.

```
package logica;

import java.awt.Image;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.io.IOException;

import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.sound.sampled.FloatControl;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.UnsupportedAudioFileException;
import javax.swing.ImageIcon;
import javax.swing.JLabel;

import interfazGrafica.EscenarioJuego;

/**
```

```
 * Declaración de clase Pajaro
 *
 * @author Julian Espinoza
 *
 */
public class Pajaro extends Hilo implements KeyListener {

	// Objects
	private JLabel pajarito;
	private EscenarioJuego juego;
	private Clip clip;

	// Attributes
	private int ejeDireccion = 1;
	private int StaticY;
	private boolean terminar;
	private boolean delay = false;
	private int alturaDePaso = 150;
	public int velocidad = 0;

	// Construct
	/**
	 * Constructor Pajaro
	 *
	 * @param pajarito
	 * @param juego
	 */
	public Pajaro(JLabel pajarito, EscenarioJuego juego) {
		this.pajarito = pajarito;
		this.juego = juego;
		this.terminar = false;
	}

	// Method for Thread
	/**
	 * Metodo que corre el movimiento del pajaro
	 */
	public void run() {
		while (true) {
			while (!terminar) {
				if (this.ejeDireccion == 1) {
					if (delay) {
						try {
							Thread.sleep(10);
						} catch (InterruptedException e1) {
						}
```

```
                }
                delay = false;

                if (this.pajarito.getY() + (velocidad / 10 * this.ejeDireccion)
                <= -150) {
                        this.pajarito.setLocation(this.pajarito.getX(), -150);
                } else if (this.pajarito.getY() + (velocidad / 10 *
                this.ejeDireccion) >= this.juego.getWindowH()  - 99 –
                this.juego.getBirdSizeH()) {
                        this.pajarito.setLocation(this.pajarito.getX(),
                        this.juego.getWindowH() - 99 –
                        this.juego.getBirdSizeH());
                } else {
                        this.pajarito.setLocation(this.pajarito.getX(),
                        this.pajarito.getY() + (velocidad / 10 *
                        this.ejeDireccion));
                }

                if (this.juego.getActivePJ() == "bird/bird") {
                        if (alturaDePaso - this.pajarito.getY() == 0) {
                                juego.bird.setIcon(new ImageIcon(juego
                                .imagenBird0.getImage().getScaledInstance
                                (juego.getBirdSizeH(),juego.getBirdSizeW() -
                                10, Image.SCALE_SMOOTH)));
                        }
                        if (this.pajarito.getY() - alturaDePaso >= 10 &
                        this.pajarito.getY() -  alturaDePaso < 30) {
                                juego.bird.setIcon(new ImageIcon(juego
                                .imagenBird20.getImage().getScaledInstance
                                (juego.getBirdSizeH(), juego.getBirdSizeW(),
                                Image.SCALE_SMOOTH)));
                        }
                        if (this.pajarito.getY() - alturaDePaso >= 30 &
                        this.pajarito.getY() -  alturaDePaso < 50) {
                                juego.bird.setIcon(new ImageIcon(juego
                                .imagenBird45.getImage().getScaledInstance
                                 (juego.getBirdSizeH(), juego.getBirdSizeW(),
                                 Image.SCALE_SMOOTH)));
                        }
                        if (this.pajarito.getY() - alturaDePaso >= 50 &
                        this.pajarito.getY() –alturaDePaso < 70) {
                                juego.bird.setIcon(new ImageIcon(juego
                                .imagenBird75.getImage().getScaledInstance
                                (juego.getBirdSizeH(), juego.getBirdSizeW(),
                                Image.SCALE_SMOOTH)));
                        }
```

```
                    if (this.pajarito.getY() - alturaDePaso >= 70) {
                        juego.bird.setIcon(new ImageIcon(juego
                        .imagenBird90.getImage().getScaledInstance
                        (juego.getBirdSizeH() - 5, juego.getBird
                        SizeW(), Image.SCALE_SMOOTH)));
                    }
                }

                if (this.juego.getActivePJ() == "space/space") {
                    if (alturaDePaso - this.pajarito.getY() == 0) {
                        juego.bird.setIcon(new ImageIcon(juego
                        .imagenSpace0.getImage().getScaled
                         Instance(juego.getBirdSizeH(),
                         juego.getBirdSizeW() - 10,
                         Image.SCALE_SMOOTH)));
                    }
                    if (this.pajarito.getY() - alturaDePaso >= 10 &
                    this.pajarito.getY() –alturaDePaso < 30) {
                        juego.bird.setIcon(new ImageIcon(juego
                        .imagenSpace20.getImage().getScaled
                         Instance(3juego.getBirdSizeH(),
                         juego.getBirdSizeW(),Image
                         .SCALE_SMOOTH)));
                    }
                    if (this.pajarito.getY() - alturaDePaso >= 30 &
                    this.pajarito.getY() – alturaDePaso < 50) {
                        juego.bird.setIcon(new ImageIcon(juego
                        .imagenSpace45.getImage().getScaled
                         Instance(juego.getBirdSizeH(),
                         juego.getBirdSizeW(), Image
                         .SCALE_SMOOTH)));
                    }
                    if (this.pajarito.getY() - alturaDePaso >= 50 &
                    this.pajarito.getY() – alturaDePaso < 70) {
                        juego.bird.setIcon(new ImageIcon(juego
                        .imagenSpace75.getImage().getScaled
                         Instance(juego.getBirdSizeH(),
                         .getBirdSizeW(), Image.SCALE_SMOOTH)));
                    }
                    if (this.pajarito.getY() - alturaDePaso >= 70) {
                        juego.bird.setIcon(new ImageIcon(juego
                        .imagenSpace90.getImage().getScaled
                         Instance(juego.getBirdSizeH() - 5,
                         juego.getBirdSizeW(), Image
                         .SCALE_SMOOTH)));
                    }
```

```
                                }

                                try {
                                        Thread.sleep(6);
                                } catch (InterruptedException e) {
                                }
                                velocidad++;
                        }
                        while (this.ejeDireccion == -1) {
                                if (this.StaticY - 45 != this.pajarito.getY()) {
                                        this.pajarito.setLocation(this.pajarito.getX(),
                                        this.pajarito.getY() + (1 * this.ejeDireccion));
                                        try {
                                                Thread.sleep(4);
                                        } catch (InterruptedException e) {
                                        }
                                } else {
                                        this.ejeDireccion = 1;
                                }
                        }
                        this.juego.update();
                }
                //
                try {
                        Thread.sleep(1);
                } catch (InterruptedException e) {
                }
        }
}

// General Methods
/**
 * Inicia la inhabilitacion del movimiento del pajaro
 */
@Override
public void pauseThread() {
        this.terminar = true;
}

/**
 * Inicia la habilitacion del movimiento del pajaro
 */
@Override
public void resumeThread() {
        this.terminar = false;
        ejeDireccion = 1;
```

```
	}

	// Methods for KeyListener
	@Override
	public void keyPressed(KeyEvent e) {
		this.ejeDireccion = -1;
		this.StaticY = this.pajarito.getY();
		if (this.juego.getActivePJ() == "bird/bird") {
			juego.bird.setIcon(new ImageIcon(juego.imagenBirdArriba.getImage()
			.getScaledInstance(juego.getBirdSizeH(),juego.getBirdSizeW(),
			Image.SCALE_SMOOTH)));
		}
		if (this.juego.getActivePJ() == "space/space") {
			juego.bird.setIcon(new ImageIcon(juego.imagenSpaceArriba.getImage()
			.getScaledInstance (juego.getBirdSizeH(),juego.getBirdSizeW(),
			Image.SCALE_SMOOTH)));
		}

		delay = true;
		alturaDePaso = this.StaticY;
		velocidad = 0;

		try {
			clip = AudioSystem.getClip();
			if (this.juego.getActivePJ() == "bird/bird") {

				clip.open(AudioSystem.getAudioInputStream(getClass()
				.getResourceAsStream("/audio/effects/jump.wav")));
			}
			if (this.juego.getActivePJ() == "porky/porky") {
				clip.open(AudioSystem.getAudioInputStream(getClass()
				.getResourceAsStream("/audio/effects/oink.wav")));
			}
			if (this.juego.getActivePJ() == "space/space") {
				clip.open(AudioSystem.getAudioInputStream(getClass()
				.getResourceAsStream("/audio/effects/spaceship.wav")));
			}

			FloatControl volumen = (FloatControl) clip.getControl(FloatControl
			.Type.MASTER_GAIN);
			volumen.setValue((float) -30.0);
			clip.loop(0);
		} catch (LineUnavailableException e1) {
		} catch (IOException e1) {
		} catch (UnsupportedAudioFileException e1) {
		}
```

```
        }

        @Override
        public void keyReleased(KeyEvent e) {
        }

        @Override
        public void keyTyped(KeyEvent e) {
        }
}
```

Código 4 Clase Pajaro.java

4.4 MovimientoTuberias.java: Clase que maneja el movimiento y el reposicionamiento de las tuberías del juego.

```
package logica;

import javax.swing.JLabel;

import interfazGrafica.EscenarioJuego;

/**
 * Declaración de clase Movimiento Tuverias
 *
 * @author Santiago Herrera
 *
 */
public class MovimientoTuberias extends Hilo {

        // Attributes
        private JLabel tuberiaAlta1, tuberiaBaja1, tuberiaAlta2, tuberiaBaja2;
        private EscenarioJuego juego;
        private int reubicacionTubos1 = 0, reubicacionTubos2 = -300;
        private boolean terminar;

        /**
         * Constructor de MovimientoTuberias
         *
         * @param tuberiaAlta1
         * @param tuberiaBaja1
         * @param tuberiaAlta2
         * @param tuberiaBaja2
         * @param juego
         */
```

```
public MovimientoTuberias(JLabel tuberiaAlta1, JLabel tuberiaBaja1, JLabel
 tuberiaAlta2, JLabel tuberiaBaja2,EscenarioJuego juego) {
        this.tuberiaAlta1 = tuberiaAlta1;
        this.tuberiaBaja1 = tuberiaBaja1;
        this.tuberiaAlta2 = tuberiaAlta2;
        this.tuberiaBaja2 = tuberiaBaja2;
        this.juego = juego;
        this.terminar = false;
}

/**
 * Método corre el movimiento de las tuberias
 */
public void run() {
        while (true) {
                while (!terminar) {
                        if (reubicacionTubos1 % 600 == 0) {
                                int altura = (int) (Math.random() * 30 - 18);
                                reubicacionTubos1 = 0;
                                this.tuberiaAlta1.setLocation(500, -350 + altura * 10);
                                this.tuberiaBaja1.setLocation(500, 370 + altura * 10);

                        }
                        if (reubicacionTubos2 % 600 == 0) {
                                reubicacionTubos2 = 0;
                                int altura = (int) (Math.random() * 30 - 18);
                                this.tuberiaAlta2.setLocation(500, -350 + altura * 10);
                                this.tuberiaBaja2.setLocation(500, 370 + altura * 10);
                        }

                        this.tuberiaAlta1.setLocation(this.tuberiaAlta1.getX() - 1,
                         this.tuberiaAlta1.getY());
                        this.tuberiaBaja1.setLocation(this.tuberiaBaja1.getX() - 1,
                         this.tuberiaBaja1.getY());
                        this.tuberiaAlta2.setLocation(this.tuberiaAlta2.getX() - 1,
                         this.tuberiaAlta2.getY());
                        this.tuberiaBaja2.setLocation(this.tuberiaBaja2.getX() - 1,
                         this.tuberiaBaja2.getY());

                        if (this.tuberiaAlta1.getX() + 100 ==
                         this.juego.getBird().getX()) {
                                this.juego.addPunto();
                        }
                        if (this.tuberiaAlta2.getX() + 100 ==
                         this.juego.getBird().getX()) {
                                this.juego.addPunto();
```

```
                        }

                        reubicacionTubos1++;
                        reubicacionTubos2++;

                        try {
                                Thread.sleep(10);
                        } catch (InterruptedException e) {
                        }

                        this.juego.update();
                }
                try {
                        Thread.sleep(1);
                } catch (InterruptedException e) {
                }
        }
    }

    /**
     * Inicia la inhabilitacion del movimiento de las tuberias
     */
    public void pauseThread() {
        this.terminar = true;
    }

    /**
     * Inicia la habilitacion del movimiento de las tuberias
     */
    public void resumeThread() {
        reubicacionTubos1 = 0;
        reubicacionTubos2 = -300;
        this.terminar = false;
    }
}
```

Código 5 Clase MovimientoTuberias.java

4.5 MovimientoTerreno.java: Clase que controla el movimiento del terreno.

```
package logica;

import javax.swing.JLabel;

import interfazGrafica.EscenarioJuego;
```

```
/**
 * Declaración de clase MovimientoTerreno
 *
 * @author Jesus Lozada
 *
 */
public class MovimientoTerreno extends Hilo {
	// Attributes
	private JLabel terreno1;
	private JLabel terreno2;
	private EscenarioJuego juego;
	private int reubicacionTerreno1 = -500;
	private int reubicacionTerreno2 = 0;
	private boolean terminar;

	/**
	 * Cronstructor Movimiento terreno
	 *
	 * @param terreno1
	 * @param terreno2
	 * @param juego
	 */
	public MovimientoTerreno(JLabel terreno1, JLabel terreno2, EscenarioJuego juego) {
		this.terreno1 = terreno1;
		this.terreno2 = terreno2;
		this.juego = juego;
		this.terminar = false;
	}

	/**
	 * Método corre el movimiento del terreno base
	 */
	public void run() {
		while (true) {
			while (!terminar) {
				if (reubicacionTerreno1 == -1000) {
					reubicacionTerreno1 = 0;
					terreno1.setLocation(500, terreno1.getY());
				}
				if (reubicacionTerreno2 == -1000) {
					reubicacionTerreno2 = 0;
					terreno2.setLocation(500, terreno2.getY());
				}

				terreno1.setLocation(terreno1.getX() - 1, terreno1.getY());
				terreno2.setLocation(terreno2.getX() - 1, terreno2.getY());
```

```
                        reubicacionTerreno1--;
                        reubicacionTerreno2--;

                        try {
                                Thread.sleep(10);
                        } catch (InterruptedException e) {
                        }

                        this.juego.update();
                }
                try {
                        Thread.sleep(1);
                } catch (InterruptedException e) {
                }
        }
    }

    /**
     * Inicia la inhabilitacion del movimiento del terreno base
     */
    public void pauseThread() {
        this.terminar = true;
    }

    /**
     * Inicia la habilitacion del movimiento del terreno base
     */
    public void resumeThread() {
        reubicacionTerreno1 = -500;
        reubicacionTerreno2 = 0;
        terreno1.setLocation(0, terreno1.getY());
        terreno2.setLocation(500, terreno2.getY());
        this.terminar = false;
    }
}
```

Código 6 Clase MovimientoTerreno.java

4.6 Colisionador.java: Clase que hereda de Hilo y controla las colisiones entre el pájaro y las tuberías/suelo y llama los métodos pertinentes cuando ocurre una colisión.

```
package logica;

import java.io.IOException;

import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.sound.sampled.FloatControl;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.UnsupportedAudioFileException;
import javax.swing.JLabel;

import interfazGrafica.EscenarioJuego;

/**
 * Declaración de clase Colisionador
 *
 * @author Jesus Lozada
 *
 */
public class Colisionador extends Hilo {
	// Attributes
	private Clip clip;

	// Components
	private JLabel bird;
	private JLabel tb1;
	private JLabel tb2;
	private JLabel tb3;
	private JLabel tb4;
	private JLabel t1;
	private JLabel t2;
	private EscenarioJuego ej;
	private boolean terminar;

	// Construct
	/**
	 * Constructor Colisionador
	 *
	 * @param bird
	 * @param tb1
	 * @param tb2
	 * @param tb3
	 * @param tb4
	 * @param ej
	 * @param t1
	 * @param t2
```

```
 */
public Colisionador(JLabel bird, JLabel tb1, JLabel tb2, JLabel tb3, JLabel tb4,
 EscenarioJuego ej, JLabel t1, JLabel t2) {
        this.bird = bird;
        this.tb1 = tb1;
        this.tb2 = tb2;
        this.tb3 = tb3;
        this.tb4 = tb4;
        this.t1 = t1;
        this.t2 = t2;
        this.ej = ej;
        this.terminar = false;
}

// Methods
/**
 * Metodo corre las condiciones actualizables de la colisiones entre el pajaro y
 * las tuberias
 */
public void run() {
        while (true) {
                while (!terminar) {
                        if (this.bird.getBounds().intersects(this.tb1.getBounds())
                                || this.bird.getBounds().intersects(this.tb2.getBounds())
                                || this.bird.getBounds().intersects(this.tb3.getBounds())
                                || this.bird.getBounds().intersects(this.tb4.getBounds())
                                || this.bird.getBounds().intersects(this.t1.getBounds())
                                || this.bird.getBounds().intersects(this.t2.getBounds())) {

                                // Reproducir sonido de golpe
                                try {
                                        clip = AudioSystem.getClip();
                                        clip.open(AudioSystem.getAudioInput
                                        Stream(getClass().getResourceAsStream
                                        ("/audio/effects/hit.wav")));

                                        FloatControl volumen = (FloatControl)
                                        clip.getControl(FloatControl.Type.MASTER_GAIN);
                                        volumen.setValue((float) -20.0);
                                        clip.loop(0);
                                } catch (LineUnavailableException e1) {
                                } catch (IOException e1) {
                                } catch (UnsupportedAudioFileException e1) {
                                }

                                this.ej.stopMusic();
```

```
                                this.ej.reset();

                                terminar = true;
                        }
                        try {
                                Thread.sleep(1);
                        } catch (InterruptedException e) {
                        }
                }
                try {
                        Thread.sleep(1);
                } catch (InterruptedException e) {
                }
        }
    }

    /**
     * Inicia la inhabilitacion del colisionador
     */
    public void resumeThread() {
        terminar = false;
    }

    @Override
    public void pauseThread() {
        // TODO Auto-generated method stub

    }
}
```

Código 7 Clase Colisionador.java

4.7 CodeListener.java: Clase que almacena lo que el jugador va ingresando por teclado con el fin de activar códigos del juego:

```
package logica;

import java.awt.Color;
import java.awt.Image;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;

import javax.swing.ImageIcon;
```

```
import interfazGrafica.EscenarioJuego;

/**
 *
 * @author Julian Espinoza
 *
 */
public class CodeListener extends Thread implements KeyListener {
	// Attributes
	private String codigo;
	private EscenarioJuego ej;

	// Construct
	public CodeListener(EscenarioJuego ej) {
		this.codigo = "";
		this.ej = ej;
	}

	// Methods
	public void run() {
		while (true) {
			if (codigo.indexOf("space") != -1) {
				if (this.ej.getActivePJ() == "space/space") {
					this.ej.getCampoJuego().remove(this.ej.getFondo());
					this.ej.setFondo(new PanelBackground("/Imagenes/
					fondo.png"));
					this.ej.getFondo().setBounds(0, 0, this.ej.getWindowW(),
					this.ej.getWindowH() - 95);
					this.ej.getCampoJuego().add(this.ej.getFondo(), new
					Integer(-1));
					this.ej.getLblPuntos().setForeground(Color.black);
					this.ej.setActivePJ("bird/bird");
					this.ej.stopMusic();
					this.ej.addMusic();

					this.ej.getTuberiaAlta1().setIcon(new ImageIcon
					(this.ej.getImagenTuboAlto().getImage()
					.getScaledInstance(100, 600, Image.SCALE_SMOOTH)));
					this.ej.getTuberiaAlta2().setIcon(new ImageIcon(this.ej
					.getImagenTuboAlto().getImage().getScaledInstance(100,
					 600, Image.SCALE_SMOOTH)));
					this.ej.getTuberiaBaja1().setIcon(new ImageIcon(this.ej
					.getImagenTuboBajo().getImage().getScaledInstance(100,
					600, Image.SCALE_SMOOTH)));
					this.ej.getTuberiaBaja2().setIcon(new ImageIcon(this.ej
					.getImagenTuboBajo().getImage().getScaledInstance(100,
```

```
                        600, Image.SCALE_SMOOTH)));
                } else {
                        this.ej.getCampoJuego().remove(this.ej.getFondo());
                        this.ej.setFondo(new PanelBackground("/Imagenes/
                        space.png"));
                        this.ej.getFondo().setBounds(0, 0, this.ej.getWindowW(),
                        this.ej.getWindowH() - 95);
                        this.ej.getCampoJuego().add(this.ej.getFondo(), new
                        Integer(-1));
                        this.ej.getLblPuntos().setForeground(Color.white);
                        this.ej.setActivePJ("space/space");
                        this.ej.stopMusic();
                        this.ej.addMusic();

                        this.ej.getTuberiaAlta1().setIcon(new ImageIcon(this.ej
                        .getImagenAsteroideAlto().getImage().getScaled
                        Instance(100, 600, Image.SCALE_SMOOTH)));
                        this.ej.getTuberiaAlta2().setIcon(new ImageIcon(this.ej
                        .getImagenAsteroideAlto().getImage().getScaled
                        Instance(100, 600, Image.SCALE_SMOOTH)));
                        this.ej.getTuberiaBaja1().setIcon(new ImageIcon(this.ej
                        .getImagenAsteroideBajo().getImage().getScaled
                        Instance(100, 600, Image.SCALE_SMOOTH)));
                        this.ej.getTuberiaBaja2().setIcon(new ImageIcon(this.ej
                        .getImagenAsteroideBajo().getImage().getScaled
                         Instance(100, 600, Image.SCALE_SMOOTH)));
                }

                this.codigo = "";
                this.ej.update();
        }
        if (codigo.indexOf("night") != -1) {
                this.ej.getCampoJuego().remove(this.ej.getFondo());
                this.ej.setFondo(new PanelBackground("/Imagenes/fondo
                 Noche.png"));
                this.ej.getFondo().setBounds(0, 0, this.ej.getWindowW(),
                 this.ej.getWindowH() - 95);
                this.ej.getCampoJuego().add(this.ej.getFondo(), new Integer(-1));
                this.ej.getLblPuntos().setForeground(Color.white);
                this.codigo = "";
                this.ej.update();
        }
        if (codigo.indexOf("day") != -1) {
                this.ej.getCampoJuego().remove(this.ej.getFondo());
                this.ej.setFondo(new PanelBackground("/Imagenes/fondo.png"));
                this.ej.getFondo().setBounds(0, 0, this.ej.getWindowW(),
```

```
                        this.ej.getWindowH() - 95);
                        this.ej.getCampoJuego().add(this.ej.getFondo(), new Integer(-1));
                        this.ej.getLblPuntos().setForeground(Color.black);
                        this.codigo = "";
                        this.ej.update();
                }
                if (codigo.indexOf("duque") != -1) {
                        if (this.ej.getActivePJ() == "porky/porky") {
                                this.ej.setActivePJ("bird/bird");

                        } else {
                                this.ej.setActivePJ("porky/porky");
                                this.ej.getBird().setIcon(new ImageIcon(this.ej.getImagen
                                Porky().getImage()
                                .getScaledInstance(this.ej.getBirdSizeH(),
                                this.ej.getBirdSizeW(),Image.SCALE_SMOOTH)));
                        }
                        this.codigo = "";
                }
                try {
                        Thread.sleep(1);
                } catch (InterruptedException e) {
                }
        }
    }

    @Override
    public void keyPressed(KeyEvent e) {
            codigo = codigo + e.getKeyChar();
    }

    @Override
    public void keyReleased(KeyEvent arg0) {
    }

    @Override
    public void keyTyped(KeyEvent arg0) {
    }
}
```

Código 8 Clase CodeListener.java

4.8 PanelBackground: Es una clase que hereda de JPanel, que permite pasarle la ruta de una imagen que colocara de fondo en el panel.

```
package logica;

import java.awt.Graphics;
import java.awt.Image;

import javax.swing.ImageIcon;
import javax.swing.JPanel;

/**
 * Declaración de clase PanelBackGround
 *
 * @author Jesus Lozada
 *
 */
public class PanelBackground extends JPanel {
	private static final long serialVersionUID = 1L;
	private Image fondo;

	/**
	 * Constructor PanelBackground
	 *
	 * @param s
	 */
	public PanelBackground(String s) {
		this.fondo = new ImageIcon(getClass().getResource(s)).getImage();
	}

	@Override
	/**
	 * @param g
	 */
	public void paint(Graphics g) {
		g.drawImage(fondo, 0, 0, getWidth(), getHeight(), this);
		this.setOpaque(false);
		super.paint(g);
	}
}
```

Código 9 Clase PanelBackground.java

5. Paquete interfazGrafica

En este paquete están las interfaces que componen el apartado visual del juego que se divide en tres clases:

5.1 EscenarioJuego.java: Esta es la interfaz principal donde están instanciados todos los elementos que componen el juego así como los métodos necesarios para su correcta actualización, se utiliza un KeyListener que controla el manejo general del juego.

```
package interfazGrafica;

import java.awt.Dimension;

import java.awt.Font;
import java.awt.Image;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.io.IOException;

import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.sound.sampled.FloatControl;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.UnsupportedAudioFileException;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JLayeredPane;
import javax.swing.SwingConstants;

import logica.CodeListener;
import logica.Colisionador;
import logica.MovimientoTerreno;
import logica.MovimientoTuberias;
import logica.Pajaro;
import logica.PanelBackground;
import modelo.Puntaje;

/**
 * Declaración de clase EscenarioJuego
 * @author Jesus Lozada.
```

```
 *
 */
public class EscenarioJuego implements KeyListener{
	// Objects
	private Pajaro pelota;
	private MovimientoTuberias tubos;
	private Colisionador col;
	private CodeListener cod;
	private MovimientoTerreno terrenos;
	private Puntaje db;
	private Clip music;

	// Components
	private JFrame frame;
	private PopUpInfo info;
	private JLabel puntajeActual;
	private JLabel mejorPuntaje;
	private JLabel tuberiaAlta1,tuberiaBaja1,tuberiaAlta2,tuberiaBaja2;
	private JLabel terreno1, terreno2;
	private JLabel lblPuntos;
	private PanelBackground fondo;
	private PanelBackground suelo;
	private PanelMenuPrincipal menuPrincipal;
	private JLayeredPane campoJuego;
	private ImageIcon imagenTuboAlto;
	private ImageIcon imagenTuboBajo;
	private ImageIcon imagenAsteroideAlto;
	private ImageIcon imagenAsteroideBajo;
	private ImageIcon terreno;

	public ImageIcon imagenBird0,imagenBirdArriba,imagenBird20,imagenBird45,
	 imagenBird90,imagenBird75;
	public ImageIcon imagenSpace0,imagenSpaceArriba,imagenSpace20,
	 imagenSpace45,imagenSpace90, imagenSpace75;
	public ImageIcon imagenPorky;
	public JLabel bird;

	// Attributes
	private int jugando = 0;
	private int puntos = 0;
	private String activePJ = "bird/bird";

	// Constants
	private final int windowH = 600;
	private final int windowW = 500;
	private final int birdSizeW = 40;
```

```
    private final int birdSizeH = 40;

    // Construct
    /**
     * Constructor de ventana principal
     * @param db
     */
    public EscenarioJuego(Puntaje db) {
        this.db = db;

        frame = new JFrame("Flappy Bird");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(windowW,windowH);
        frame.setLocationRelativeTo(null);
        frame.setUndecorated(true);

        imagenBird0 = new ImageIcon(getClass().getResource("/Imagenes
        /bird/bird.png"));
        imagenBirdArriba = new ImageIcon(getClass().getResource("/Imagenes/
         bird/bird20.png"));
        imagenBird20 = new ImageIcon(getClass().getResource("/Imagenes/bird/bird –
        20.png"));
        imagenBird45 = new ImageIcon(getClass().getResource("/Imagenes/bird/bird –
        45.png"));
        imagenBird75 = new ImageIcon(getClass().getResource("/Imagenes/bird/bird –
        75.png"));
        imagenBird90 = new ImageIcon(getClass().getResource("/Imagenes/bird/bird –
        90.png"));

        imagenSpace0 = new ImageIcon(getClass().getResource("/Imagenes/space/
        space 0.png"));
        imagenSpaceArriba = new ImageIcon(getClass().getResource("/Imagenes/
        space/space 20.png"));
        imagenSpace20 = new ImageIcon(getClass().getResource("/Imagenes/
        space/space -20.png"));
        imagenSpace45 = new ImageIcon(getClass().getResource("/Imagenes/
        space/space -45.png"));
        imagenSpace75 = new ImageIcon(getClass().getResource("/Imagenes/
        space/space -75.png"));
        imagenSpace90 = new ImageIcon(getClass().getResource("/Imagenes/
        space/space -90.png"));

        imagenPorky = new ImageIcon(getClass().getResource("/Imagenes/
         porky/porky.png"));

        campoJuego = new JLayeredPane();
```

```
campoJuego.setPreferredSize(new Dimension(this.windowW,this.windowH));
campoJuego.setLayout(null);

//Principal menu
fondo = new PanelBackground("/Imagenes/fondo.png");
fondo.setBounds(0, 0,this.windowW, this.windowH-95);
campoJuego.add(fondo, new Integer(-1));

info = new PopUpInfo(this);
info.setVisible(false);

menuPrincipal = new PanelMenuPrincipal(this);
menuPrincipal.setLocation(50,120);
campoJuego.add(menuPrincipal);

mejorPuntaje = new JLabel();
if(this.db.getPuntajes().size() < 1) {
        this.mejorPuntaje.setText("Best: 0");
} else {
        this.mejorPuntaje.setText("Best: " + this.db.getPuntajes().get(0));
}
mejorPuntaje.setHorizontalAlignment(SwingConstants.CENTER);
mejorPuntaje.setFont(new Font("Agency FB", mejorPuntaje.getFont()
.getStyle(), 70));
menuPrincipal.getPaneles()[1].add(mejorPuntaje);

puntajeActual = new JLabel();
puntajeActual.setFont(new Font("Agency FB", puntajeActual.getFont()
.getStyle(), 30));
puntajeActual.setHorizontalAlignment(SwingConstants.CENTER);
menuPrincipal.getPaneles()[1].add(puntajeActual);

//Game
suelo = new PanelBackground("/Imagenes/suelo.png");
suelo.setBounds(0, this.windowH-100,this.windowW, 100);
campoJuego.add(suelo, new Integer(1));

imagenTuboAlto = new ImageIcon(getClass().getResource("/Imagenes/
tuboArriba.png"));
imagenTuboBajo = new ImageIcon(getClass().getResource("/Imagenes/
tuboAbajo.png"));
imagenAsteroideAlto = new ImageIcon(getClass().getResource("/Imagenes/
asteroideArriba.png"));
imagenAsteroideBajo = new ImageIcon(getClass().getResource("/Imagenes/
asteroideAbajo.png"));
terreno = new ImageIcon(getClass().getResource("/Imagenes/div.png"));
```

```
bird = new JLabel();
bird.setBounds(100, (this.windowH/2)-100-(this.birdSizeH/2),
this.birdSizeW,this.birdSizeH);
bird.setIcon(new ImageIcon(imagenBird0.getImage().getScaledInstance
(this.birdSizeH, this.birdSizeW-10, Image.SCALE_SMOOTH)));
campoJuego.add(bird, new Integer(0));
bird.setVisible(false);

tuberiaAlta1 = new JLabel();
tuberiaAlta1.setBounds(500, -350, 100, 600);
tuberiaAlta1.setIcon(new ImageIcon(imagenTuboAlto.getImage()
.getScaledInstance(100, 600, Image.SCALE_SMOOTH)));
campoJuego.add(tuberiaAlta1, new Integer(0));

tuberiaBaja1 = new JLabel();
tuberiaBaja1.setBounds(500,370, 100, 600);
tuberiaBaja1.setIcon(new ImageIcon(imagenTuboBajo.getImage()
.getScaledInstance(100, 600, Image.SCALE_SMOOTH)));
campoJuego.add(tuberiaBaja1, new Integer(0));

tuberiaAlta2 = new JLabel();
tuberiaAlta2.setBounds(850, -350, 100, 600);
tuberiaAlta2.setIcon(new ImageIcon(imagenTuboAlto.getImage()
.getScaledInstance(100, 600, Image.SCALE_SMOOTH)));
campoJuego.add(tuberiaAlta2, new Integer(0));

tuberiaBaja2 = new JLabel();
tuberiaBaja2.setBounds(800,370, 100, 600);
tuberiaBaja2.setIcon(new ImageIcon(imagenTuboBajo.getImage()
.getScaledInstance(100, 600, Image.SCALE_SMOOTH)));
campoJuego.add(tuberiaBaja2, new Integer(0));

terreno1 = new JLabel();
terreno1.setBounds(0, windowH-100, this.windowW, 20);
terreno1.setIcon(new ImageIcon(terreno.getImage()));
campoJuego.add(terreno1, new Integer(2));

terreno2 = new JLabel();
terreno2.setBounds(500, windowH-100, this.windowW, 20);
terreno2.setIcon(new ImageIcon(terreno.getImage()));
campoJuego.add(terreno2, new Integer(2));

lblPuntos = new JLabel("Score: " + this.puntos);
lblPuntos.setBounds(15, 10, this.windowW, 20);
campoJuego.add(lblPuntos, new Integer(2));
```

```
        lblPuntos.setVisible(false);

        pelota = new Pajaro(bird, this);
        tubos = new MovimientoTuberias(tuberiaAlta1,tuberiaBaja1,tuberiaAlta2,
        tuberiaBaja2,this);
        col = new Colisionador(bird,tuberiaBaja1,tuberiaAlta1,tuberiaAlta2,
        tuberiaBaja2, this, terreno1, terreno2);
        terrenos = new MovimientoTerreno(terreno1, terreno2, this);
        cod = new CodeListener(this);

        pelota.start();
        pelota.pauseThread();
        tubos.start();
        tubos.pauseThread();
        terrenos.start();
        terrenos.pauseThread();
        col.start();
        cod.start();

        frame.addKeyListener(this);

        frame.add(campoJuego);
        frame.setSize(this.windowW, windowH);
        frame.setVisible(true);
        frame.setResizable(false);

        this.update();
}

// General Methods
/**
 * Método repinta la pantalla
 */
public void update() {
        lblPuntos.setText("Score: " + this.puntos);
        puntajeActual.setText("Score: "+ this.puntos);
        frame.getContentPane().repaint();
}
/**
 * Método reinicia el juego
 */
public void reset() {

        this.db.addPuntos(this.puntos);
        this.db.ordenarPuntos();
```

```
            if(this.db.getPuntajes().size() < 1) {
                this.mejorPuntaje.setText("Best: 0");
            } else {
                this.mejorPuntaje.setText("Best: " + this.db.getPuntajes().get(0));
            }

            frame.removeKeyListener(pelota);
            frame.removeKeyListener(cod);
            tubos.pauseThread();
            terrenos.pauseThread();
            menuPrincipal.setVisible(true);

            try {
                Thread.sleep(500);
            } catch (InterruptedException e) {}

            try {
                music = AudioSystem.getClip();
                music.open(AudioSystem.getAudioInputStream(getClass().getResource
                AsStream("/audio/effects/game over.wav")));
                FloatControl volumen = (FloatControl) music.getControl(FloatControl
                .Type.MASTER_GAIN);
                volumen.setValue((float) -30.0);
                music.loop(0);
            } catch (LineUnavailableException e1) {
            } catch (IOException e1) {
            } catch (UnsupportedAudioFileException e1) {
            }

            while(bird.getY() <= windowH-100-birdSizeH) {
                try {
                    Thread.sleep(10);
                } catch (InterruptedException e) {}
            }

            pelota.pauseThread();

            this.jugando = 0;
        }
        /**
         * Método suma un punto
         */
        public void addPunto() {
            this.puntos++;
        }
        /**
```

```
	 * Método reinicia el puntaje
	 */
	public void resetPuntos() {
		this.puntos = 0;
	}
	/**
	 * Método añade música al juego
	 */
	public void addMusic() {
		try {
			music = AudioSystem.getClip();
			if(this.activePJ != "space/space") {
				music.open(AudioSystem.getAudioInputStream(getClass()
				 .getResource("/audio/Childs Nightmare.wav")));
			} else {
				music.open(AudioSystem.getAudioInputStream(getClass()
				 .getResource("/audio/Interplanetary Odyssey.wav")));
			}

			//Volumen
			FloatControl volumen = (FloatControl) music.getControl(FloatControl
			.Type.MASTER_GAIN);
			volumen.setValue((float) -35.0);

			music.loop(Clip.LOOP_CONTINUOUSLY);
		} catch (LineUnavailableException e) {} catch (IOException e) {
		} catch (UnsupportedAudioFileException e) {
		}
	}

	public void stopMusic() {
		music.stop();
	}

	// Methods for KeyListener

	public void keyPressed(KeyEvent e) {

		if(e.getKeyCode() == KeyEvent.VK_ESCAPE) {

			frame.setVisible(false);
			frame.dispose();
			System.exit(0);

		}
```

```
if(e.getKeyCode() == KeyEvent.VK_I && this.jugando == 0) {

        info.setVisible(true);

} else {

        if(this.jugando == 1) {
                pelota.velocidad = 0;
                pelota.resumeThread();
                tubos.resumeThread();
                this.update();
                this.jugando = 2;
        }
        if(this.jugando == 0) {
                this.resetPuntos();

                if(this.activePJ == "bird/bird") {
                        bird.setIcon(new ImageIcon(imagenBird0.getImage()
                        .getScaledInstance(this.birdSizeH, this.birdSizeW-10,
                        Image.SCALE_SMOOTH)));
                }
                if(this.activePJ == "space/space") {
                        bird.setIcon(new ImageIcon(imagenSpace0.getImage()
                        .getScaledInstance(this.birdSizeH, this.birdSizeW-10,
                        Image.SCALE_SMOOTH)));
                }

                this.tuberiaAlta1.setLocation(500, -350);
                this.tuberiaBaja1.setLocation(500,370);
                this.tuberiaAlta2.setLocation(850, -350);
                this.tuberiaBaja2.setLocation(800,370);
                this.bird.setLocation(100, (this.windowH/2)-100 +
                (this.birdSizeH/2));
                this.update();

                pelota.velocidad = 0;

                bird.setVisible(true);
                menuPrincipal.setVisible(false);
                lblPuntos.setVisible(true);

                this.db.addPuntos(this.puntos);
                this.db.ordenarPuntos();

                this.resetPuntos();
```

```
                        if(this.db.getPuntajes().size() < 1) {
                                this.mejorPuntaje.setText("Best: 0");
                        } else {
                                this.mejorPuntaje.setText("Best: " + this.db
                                .getPuntajes().get(0));
                        }

                        frame.addKeyListener(pelota);
                        frame.addKeyListener(cod);


                        col.resumeThread();
                        terrenos.resumeThread();

                        this.jugando = 1;
                        this.update();
                        this.addMusic();
                }
        }
}

@Override
public void keyReleased(KeyEvent e) {

}

@Override
public void keyTyped(KeyEvent e) {

}

// Getters
public JFrame getFrame() {
        return frame;
}

public int getWindowH() {
        return windowH;
}

public int getWindowW() {
        return windowW;
}

public int getBirdSizeW() {
        return birdSizeW;
```

```
	}

	public int getBirdSizeH() {
		return birdSizeH;
	}

	public JLayeredPane getCampoBola() {
		return campoJuego;
	}

	public JLayeredPane getCampoJuego() {
		return campoJuego;
	}

	public PanelBackground getFondo() {
		return fondo;
	}

	public void setFondo(PanelBackground fondo) {
		this.fondo = fondo;
	}

	public Pajaro getPelota() {
		return pelota;
	}

	public MovimientoTuberias getTubos() {
		return tubos;
	}

	public JLabel getBird() {
		return bird;
	}

	public void setActivePJ(String activePJ) {
		this.activePJ = activePJ;
	}

	public ImageIcon getImagenPorky() {
		return imagenPorky;
	}

	public String getActivePJ() {
		return activePJ;
	}
```

```
	public JLabel getLblPuntos() {
		return lblPuntos;
	}

	public JLabel getTuberiaAlta1() {
		return tuberiaAlta1;
	}

	public JLabel getTuberiaBaja1() {
		return tuberiaBaja1;
	}

	public JLabel getTuberiaAlta2() {
		return tuberiaAlta2;
	}

	public JLabel getTuberiaBaja2() {
		return tuberiaBaja2;
	}

	public ImageIcon getImagenAsteroideAlto() {
		return imagenAsteroideAlto;
	}

	public ImageIcon getImagenAsteroideBajo() {
		return imagenAsteroideBajo;
	}

	public ImageIcon getImagenTuboAlto() {
		return imagenTuboAlto;
	}

	public ImageIcon getImagenTuboBajo() {
		return imagenTuboBajo;
	}

}
```

Código 10 EscenarioJuego.java

5.2 PanelMenuPrincipal.java: Esta clase es la que contiene los elementos gráficos del menú principal del juego.

```
package interfazGrafica;
```

```
import java.awt.AlphaComposite;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.GridLayout;
import java.awt.RenderingHints;

import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JPanel;

/**
 *
 * @author Julian Espinoza
 *
 */
@SuppressWarnings("serial")
public class PanelMenuPrincipal extends JPanel {
	// Components
	private JLabel titulo;
	private JPanel[] paneles = new JPanel[3];

	// Construct
	public PanelMenuPrincipal(EscenarioJuego ej) {

		setLayout(new GridLayout(3, 1));
		setSize(400, 350);
		setOpaque(false);

		titulo = new JLabel(new ImageIcon(getClass().getResource
		("/Imagenes/Logo.png")));

		paneles[0] = new JPanel() {
			protected void paintComponent(Graphics g) {
				Graphics2D g2 = (Graphics2D) g;
				g2.setRenderingHint(RenderingHints.KEY_INTERPOLATION,
				 RenderingHints.VALUE_INTERPOLATION_BILINEAR);
				AlphaComposite old = (AlphaComposite) g2.getComposite();
				g2.setComposite(AlphaComposite.SrcOver.derive(0.7f));
				super.paintComponent(g);
				g2.setComposite(old);
			}
		};
		paneles[0].setBackground(Color.WHITE);
		paneles[0].add(titulo);
		add(paneles[0]);
```

```
        paneles[1] = new JPanel(new GridLayout(2, 1)) {
                protected void paintComponent(Graphics g) {
                        Graphics2D g2 = (Graphics2D) g;
                        g2.setRenderingHint(RenderingHints.KEY_INTERPOLATION,
                         RenderingHintsVALUE_INTERPOLATION_BILINEAR);
                        AlphaComposite old = (AlphaComposite) g2.getComposite();
                        g2.setComposite(AlphaComposite.SrcOver.derive(0.7f));
                        super.paintComponent(g);
                        g2.setComposite(old);
                }
        };
        paneles[1].setBackground(Color.WHITE);
        add(paneles[1]);

        JLabel aux = new JLabel(new ImageIcon(getClass().getResource
        ("/Imagenes/texto.png")));
        aux.setVisible(true);

        paneles[2] = new JPanel() {
                protected void paintComponent(Graphics g) {
                        Graphics2D g2 = (Graphics2D) g;
                        g2.setRenderingHint(RenderingHints.KEY_INTERPOLATION,
                         RenderingHints.VALUE_INTERPOLATION_BILINEAR);
                        AlphaComposite old = (AlphaComposite) g2.getComposite();
                        g2.setComposite(AlphaComposite.SrcOver.derive(0.7f));
                        super.paintComponent(g);
                        g2.setComposite(old);
                }
        };
        paneles[2].setBackground(Color.WHITE);
        paneles[2].setSize(500, 70);
        paneles[2].add(aux);
        add(paneles[2]);

        // Do visible
        titulo.setVisible(true);
        paneles[0].setVisible(true);
        aux.setVisible(true);
        paneles[1].setVisible(true);
        paneles[2].setVisible(true);

}

// Getters
public JPanel[] getPaneles() {
```

```
                return paneles;
        }
}
```

Código 11 Clase PanelMenuPrincipal.java

5.3 PopUpInfo.java: esta clase contiene al JFrame asociado a la ventana que muestra al usuario las instrucciones del juego como la ventana de códigos que están disponibles.

```
package interfazGrafica;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTabbedPane;
import javax.swing.SpringLayout;

/**
 * Declaración clase PopUpInfo
 *
 * @author Julian Espinoza
 *
 */
@SuppressWarnings("serial")
public class PopUpInfo extends JFrame {

        // Components
        private JPanel manualDeUsuario;
        private JPanel listaCodes;
        private JButton cerrar;

        // Constants
        private final int windowW = 400;
        private final int windowH = 300;
```

```
// Construct
/**
 * Constructor PopUpInfo
 *
 * @param ej
 */
public PopUpInfo(EscenarioJuego ej) {

	super();

	setLayout(new BorderLayout());
	setSize(this.windowW, this.windowH);
	setLocationRelativeTo(ej.getFrame());
	setUndecorated(true);

	JLabel titulo = new JLabel("Información");
	titulo.setFont(new Font(titulo.getFont().getFontName(),
	titulo.getFont().getStyle(), 17));

	cerrar = new JButton(new ImageIcon(getClass().getResource
	("/Imagenes/equis.png")));
	cerrar.setSize(25, 25);
	cerrar.setOpaque(false);
	cerrar.setContentAreaFilled(false);
	cerrar.setBorderPainted(false);
	cerrar.addActionListener(new ActionListener() {
		public void actionPerformed(ActionEvent e) {

			setVisible(false);

		}
	});

	SpringLayout spring = new SpringLayout();

	JPanel opcsVentana = new JPanel(spring);
	opcsVentana.setPreferredSize(new Dimension(this.windowW, 40));
	opcsVentana.setBackground(Color.GREEN.brighter());
	opcsVentana.setBorder(BorderFactory.createLineBorder
	(Color.GREEN.darker(), 3));
	opcsVentana.add(cerrar);
	cerrar.setVisible(true);
	opcsVentana.add(titulo);
	titulo.setVisible(true);
	spring.putConstraint(SpringLayout.WEST, cerrar, -50, SpringLayout.EAST,
```

```
                opcsVentana);
                spring.putConstraint(SpringLayout.WEST, titulo, 20, SpringLayout.WEST,
                opcsVentana);
                spring.putConstraint(SpringLayout.NORTH, titulo, 5, SpringLayout.NORTH,
                opcsVentana);
                add(opcsVentana, BorderLayout.NORTH);
                opcsVentana.setVisible(true);

                JLabel imagenManual = new JLabel(new ImageIcon(getClass().getResource
                ("/Imagenes/Logo.png")));

                JLabel imagenCodes = new JLabel(new ImageIcon(getClass().getResource
                ("/Imagenes/Logo.png")));

                manualDeUsuario = new JPanel();
                manualDeUsuario.setVisible(true);
                manualDeUsuario.add(imagenManual);
                imagenManual.setVisible(true);

                listaCodes = new JPanel();
                listaCodes.setVisible(true);
                listaCodes.add(imagenCodes);
                imagenCodes.setVisible(true);

                JTabbedPane pane = new JTabbedPane();
                pane.setBackground(Color.GREEN.brighter());
                pane.setBorder(BorderFactory.createLineBorder(Color.GREEN.darker(), 3));
                pane.addTab("Manual de Usuario", manualDeUsuario);
                pane.addTab("Codigos", listaCodes);
                add(pane, BorderLayout.CENTER);

        }

}
```

Código 12 Clase PopUpInfo.java