



TECNOLOGICO NACIONAL DE MEXICO

Instituto Tecnológico de la Laguna
Ingeniería en Sistemas Computacionales



TOPICOS AVANZADOS DE PROGRAMACION

PERIODO: Ago - Dic / 2020

GRUPO: "B" 17 – 18 Hrs

PRACTICA No. __U4P01__

Aplicaciones MultiHilo

ALUMNO:

17130800 Félix Gerardo Martínez Hinojo

PROFESOR:

Ing. Luis Fernando Gil Vázquez

Torreón, Coah. a 20 de Enero de 2021

Situación didáctica | Ejercicio

Aplicaciones visuales que permitan valerse de la clase Thread, para manejar hilos dentro de la aplicación, de tal manera que se puedan tener ciertas acciones ejecutándose intercaladamente, y cada una terminando en el tiempo que el procesador lo dicte, según el equipo de la persona que esté ejecutando las aplicaciones aquí mostradas.

Analisis

Ejercicio 1

Desarrollar una aplicación visual Java basada en hilos que simule una carrera de 3 contrincantes. Cada contrincante se debe representar en la IU con un componente de barra de progreso, dicho componente se actualizará desde un hilo que se ejecutará en segundo plano, de tal manera que cada contrincante es un hilo. Se requiere crear dos versiones de la clase que represente el hilo: una versión extendiéndola de Thread y otra versión implementando el interface Runnable. Ambas clases deben ser intercambiables, es decir la aplicación podría usar una u otra versión sin ningún otro cambio más que cambiar el nombre del paquete donde se encuentra la clase.

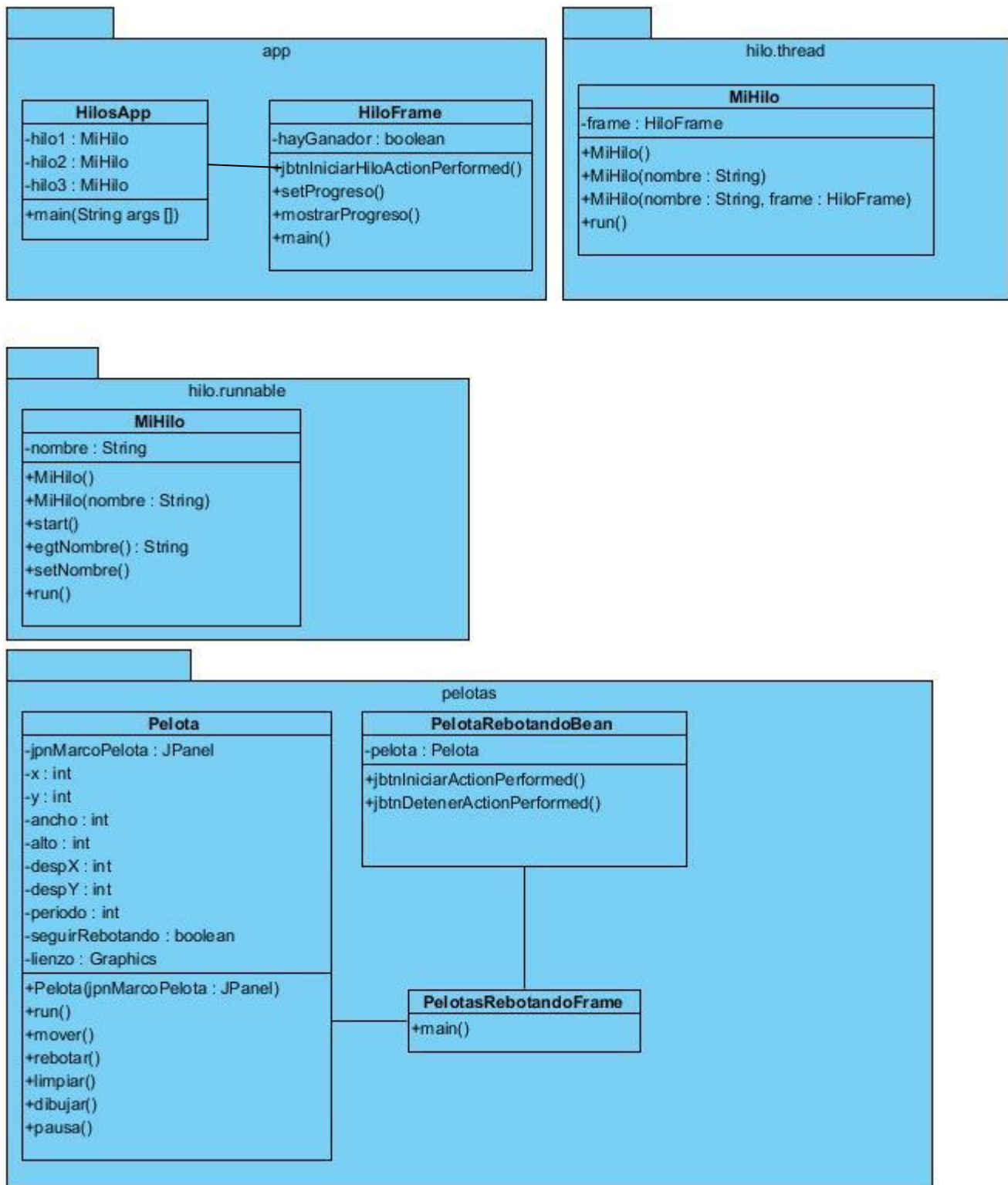
Ejercicio 2

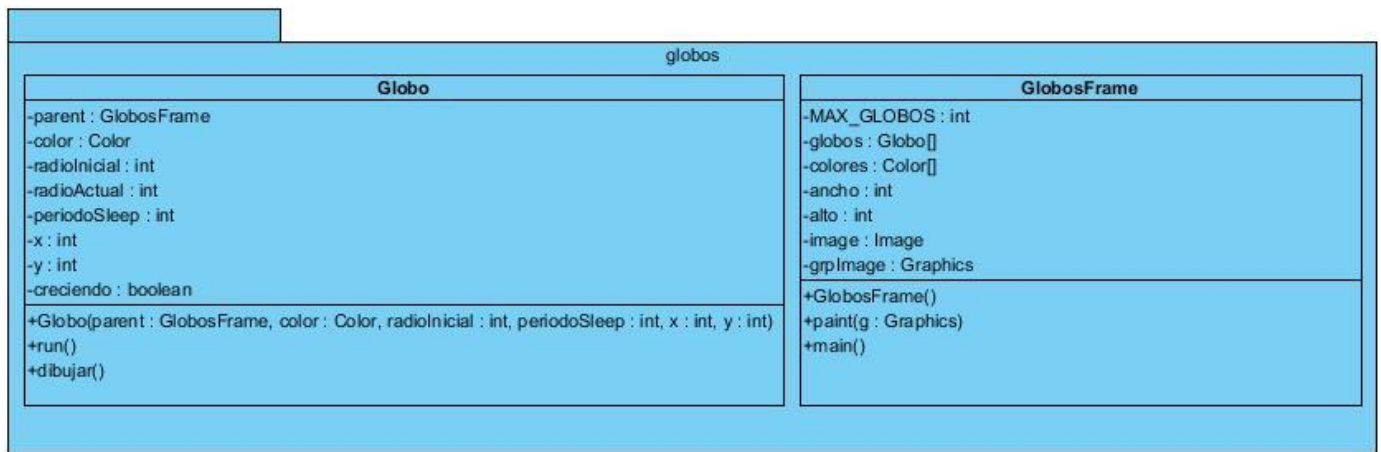
Desarrollar un JavaBean que muestre una pelotita desplazándose en un marco y rebote conforme alcanza los bordes del marco, además debe contar con dos botones uno para iniciar el desplazamiento de la pelota y otro para detenerlo. La implementación de la funcionalidad de la pelota debe ser mediante un hilo que actualice continuamente la posición de la pelota. Luego crear una aplicación visual Java donde coloque 4 instancias de este bean de tal manera que se tendrán hasta 4 pelotas rebotando al mismo tiempo.

Ejercicio 3

Desarrollar una aplicación visual Java que muestre en una ventana 10 globos en animación inflándose y desinflándose. Cada globo crece y decrece a diferentes tamaños. Cada globo debe implementarse como un hilo. Para evitar el efecto de parpadeo en pantalla por la constante actualización de los globos se debe aplicar la técnica de doble-buffer.

Diseño





Código

Ejercicio 1

MiHilo.java - Thread

```

/*-----
*:
*:          TECNOLOGICO NACIONAL DE MEXICO
*:          INSTITUTO TECNOLOGICO DE LA LAGUNA
*:          INGENIERIA EN SISTEMAS COMPUTACIONALES
*:          TOPICOS AVANZADOS DE PROGRAMACION "B"
*:
*:          SEMESTRE: AGO-DIC/2020      HORA: 17-18 HRS
*:
*:          Clase con la parte lógica programada de la carrera mediante Hilos, la clase
Thread.
*:
*: Archivo      : HiloFrame.java
*: Autor       : Félix Gerardo Martínez Hinojo      17130800
*: Fecha       : 11/ENE/2021
*: Compilador  : JAVA J2SE v1.8.0
*: Descripción : Clase con la parte lógica programada de la Carrera mediante Hilos.
*:
*: Ultima modif:
*: Fecha      Modificó      Motivo
*:-----
*: 11/ENE/2020 Félix Mtz      Se añadió el prologo.
*:-----*/
package hilo.thread;

import app.HiloFrame;
import java.util.logging.Level;
import java.util.logging.Logger;

// -----
public class MiHilo extends Thread {

    private HiloFrame frame;

    // Constructor de la clase MiHilo.
    public MiHilo() {
        super();
    }
    // -----
    // constructor MiHilo que setea el nombre del tío en carrera.
    public MiHilo ( String nombre ) {
        super ( nombre );
    }
    // -----
    // constructor con el nombre, y con el contexto del frame HiloFrame.
    public MiHilo ( String nombre, HiloFrame frame ) {
        super ( nombre );
        this.frame = frame;
    }
  
```

```

}
// -----
// Método run () de la clase Thread que pone en marcha el ( los ) Hilo(s).
@Override
public void run () {
    for (int i = 0; i <= 100; i++) {
        System.out.println( getName() + " = " + i );
        if ( frame != null )
            frame.setProgreso ( getName (), i );

        try {
            Thread.sleep ( 50 );
        } catch ( InterruptedException ex ) {
            Logger.getLogger( MiHilo.class.getName() ).log( Level.SEVERE, null, ex );
        }
    }
}
}

```

MiHilo.java - Runnable

```

/*-----
*:          TECNOLOGICO NACIONAL DE MEXICO
*:          INSTITUTO TECNOLOGICO DE LA LAGUNA
*:          INGENIERIA EN SISTEMAS COMPUTACIONALES
*:          TOPICOS AVANZADOS DE PROGRAMACION "B"
*:
*:          SEMESTRE: AGO-DIC/2020   HORA: 17-18 HRS
*:
*:          Clase con la parte lógica programada de la carrera mediante Hilos, implementando
*:          la interface Runnable.
*:
*: Archivo   : HiloFrame.java
*: Autor    : Félix Gerardo Martínez Hinojo   17130800
*: Fecha     : 11/ENE/2021
*: Compilador : JAVA J2SE v1.8.0
*: Descripción : Clase con la parte lógica programada de la Carrera mediante Hilos, con la interface
*:              runnable.
*:
*: Última modif:
*: Fecha      Modific  Motivo
*: =====
==
*: 11/ENE/2020 Félix Mtz   Se añadió el prologo.
*: -----*/
package hilo.runnable;

import app.HiloFrame;
import java.awt.Frame;
import java.util.logging.Level;
import java.util.logging.Logger;

// -----
public class MiHilo implements Runnable {

    private String nombre = "";

    // Constructor por default de la clase MiHilo.
    public MiHilo() {

    }
}

```

```
// -----
// constructor de la clase MiHilo con el nombre del tío en carrera.
public MiHilo ( String nombre ) {
    this.nombre = nombre;
}
// -----
// Método start () que permite simular el comportamiento del método start () de la clase Thread.
public void start() {
    Thread thread = new Thread ( this );
    thread.start ();
}
// -----
// Método getter para obtener el nombre del tío en carrera.
public String getNombre() {
    return nombre;
}
// -----
// Método setter para obtener el nombre del tío en carrera.
public void setNombre(String nombre) {
    this.nombre = nombre;
}
// -----
// Método run () que ejecuta el Hilo.
@Override
public void run() {
    for (int i = 0; i <= 100; i++) {
        System.out.println( getNombre () + " = " + i );

        try {
            Thread.sleep ( 50 );
        } catch ( InterruptedException ex ) {
            Logger.getLogger ( MiHilo.class.getName() ).log( Level.SEVERE, null, ex );
        }
    }
}
}
```

HilosApp.java

```
/*-----
.*          TECNOLOGICO NACIONAL DE MEXICO
.*          INSTITUTO TECNOLOGICO DE LA LAGUNA
.*          INGENIERIA EN SISTEMAS COMPUTACIONALES
.*          TOPICOS AVANZADOS DE PROGRAMACION "B"
.*
.*          SEMESTRE: AGO-DIC/2020   HORA: 17-18 HRS
.*
.*          Clase con la llamada a la parte lógica de los Hilos.
.*
.* Archivo   : HiloFrame.java
.* Autor     : Félix Gerardo Martínez Hinojo   17130800
.* Fecha     : 11/ENE/2021
.* Compilador : JAVA J2SE v1.8.0
```

```
/* Descripción : Frame con la parte visual programada de la Carrera mediante Hilos.
```

```
/*
```

```
/* Última modificación:
```

```
/* Fecha      Modificación      Motivo
```

```
/* =====
```

```
/* 11/ENE/2020 Félix Mtz      Se añadió el prologo.
```

```
/* -----*/
```

```
package app;
```

```
import hilo.runnable.MiHilo;
```

```
// -----
```

```
public class HilosApp {
```

```
    public static void main(String[] args) {
```

```
        MiHilo hilo1 = new MiHilo ( "Tio Rios" );
```

```
        MiHilo hilo2 = new MiHilo ( "Tio Oswi" );
```

```
        MiHilo hilo3 = new MiHilo ( "Ing. Ulloa" );
```

```
        hilo1.start();
```

```
        hilo2.start();
```

```
        hilo3.start();
```

```
    }
```

```
// -----
```

```
}
```

HiloFrame.java

```
/* -----
```

```
/*          TECNOLOGICO NACIONAL DE MEXICO  
/*          INSTITUTO TECNOLOGICO DE LA LAGUNA  
/*          INGENIERIA EN SISTEMAS COMPUTACIONALES  
/*          TOPICOS AVANZADOS DE PROGRAMACION "B"
```

```
/*  
/*          SEMESTRE: AGO-DIC/2020   HORA: 17-18 HRS
```

```
/*  
/*          Frame que nos muestra barras de progreso con una carrera entre 3 tíos  
/*          mediante Hilos.
```

```
/*
```

```
/* Archivo      : HiloFrame.java
```

```
/* Autor        : Félix Gerardo Martínez Hinojo   17130800
```

```
/* Fecha        : 11/ENE/2021
```

```
/* Compilador   : JAVA J2SE v1.8.0
```

```
/* Descripción  : Frame con la parte visual programada de la Carrera mediante Hilos.
```

```
/*
```

```
/* Última modificación:
```

```

:* Fecha      Modificación      Motivo
:*=====
=====
:* 11/ENE/2020 Félix Mtz      Se añadió el prologo.
:*-----*/
package app;

import hilo.thread.MiHilo;
import javax.swing.JLabel;
import javax.swing.JProgressBar;

// -----
public class HiloFrame extends javax.swing.JFrame {

    // variable booleana que permite evaluar si hay un ganador o no.
    private boolean hayGanador;

    public HiloFrame() {
        initComponents();

        lblGanador1.setVisible ( false );
        lblGanador2.setVisible ( false );
        lblGanador3.setVisible ( false );
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jTextField2 = new javax.swing.JTextField();
        jTextField3 = new javax.swing.JTextField();
        jProgressBar1 = new javax.swing.JProgressBar();
        jProgressBar2 = new javax.swing.JProgressBar();
        jProgressBar3 = new javax.swing.JProgressBar();
        lblGanador1 = new javax.swing.JLabel();
        lblGanador2 = new javax.swing.JLabel();
        lblGanador3 = new javax.swing.JLabel();
        btnIniciarHilo = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setText("1");

        jLabel2.setText("2");

```



```
.addGroup(layout.createSequentialGroup()  
    .addComponent(jLabel3)  
    .addGap(18, 18, 18)  
    .addComponent(jtxfHilo3)))  
.addGap(32, 32, 32)  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addComponent(jpbrHilo2, javax.swing.GroupLayout.PREFERRED_SIZE, 348,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
    .addComponent(jpbrHilo1, javax.swing.GroupLayout.PREFERRED_SIZE, 348,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
    .addComponent(jpbrHilo3, javax.swing.GroupLayout.PREFERRED_SIZE, 348,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
.addGap(87, 87, 87)  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,  
false)  
    .addComponent(jlblGanador3, javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
    .addComponent(jlblGanador2, javax.swing.GroupLayout.DEFAULT_SIZE, 104,  
Short.MAX_VALUE)  
    .addComponent(jlblGanador1, javax.swing.GroupLayout.Alignment.TRAILING,  
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,  
Short.MAX_VALUE))  
.addContainerGap(25, Short.MAX_VALUE))  
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,  
layout.createSequentialGroup()  
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
    .addComponent(jbtnIniciarHilo)  
    .addGap(355, 355, 355))  
);  
layout.setVerticalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
    .addGroup(layout.createSequentialGroup()  
        .addGap(67, 67, 67)  
  
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  
            .addComponent(jLabel1)  
            .addComponent(jtxfHilo1, javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)  
            .addComponent(jpbrHilo1, javax.swing.GroupLayout.PREFERRED_SIZE, 20,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
            .addComponent(jlblGanador1, javax.swing.GroupLayout.PREFERRED_SIZE, 20,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
            .addGroup(layout.createSequentialGroup()  
                .addGap(44, 44, 44)  
  
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  
                    .addComponent(jLabel2)
```

```

        .addComponent(jtxfHilo2,          javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jpbrHilo2,  javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jlblGanador2, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(38, 38, 38)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel3)
        .addComponent(jtxfHilo3,          javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jpbrHilo3,  javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(jlblGanador3,  javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(55, 55, 55)
        .addComponent(jbtnIniciarHilo)
        .addContainerGap(77, Short.MAX_VALUE)
);

pack();
} // </editor-fold>
// -----
// evento click del botón iniciar del frame visual.
private void jbtnIniciarHiloActionPerformed(java.awt.event.ActionEvent evt) {
    lblGanador1.setVisible ( false );
    lblGanador2.setVisible ( false );
    lblGanador3.setVisible ( false );

    hayGanador = false;

    MiHilo hilo1 = new MiHilo ( "Tio Rios", this );
    MiHilo hilo2 = new MiHilo ( "Tio Oswi", this );
    MiHilo hilo3 = new MiHilo ( "Tio Ulloita", this );

    hilo1.start();
    hilo2.start();
    hilo3.start();
}
// -----
// Método setProgreso () que establece el valor de los progressBar.
synchronized public void setProgreso ( String nombre, int progreso ) {

```

```

        if ( nombre.equals ( jtxfHilo1.getText () ) ) {
            mostrarProgreso ( progreso, jpbrHilo1, jlblGanador1 );
        } else if ( nombre.equals ( jtxfHilo2.getText () ) ) {
            mostrarProgreso ( progreso, jpbrHilo2, jlblGanador2 );
        } else if ( nombre.equals ( jtxfHilo3.getText () ) ) {
            mostrarProgreso ( progreso, jpbrHilo3, jlblGanador3 );
        }
    }
}
// -----
// Método mostrarProgreso () que cambia el valor de la barra de progreso dependiendo de la
// ejecución
// del Hilo.
private void mostrarProgreso ( int progreso, JProgressBar jpbrProgreso,
                               JLabel jlblGanador ) {
    jpbrProgreso.setValue( progreso );
    if ( progreso == 100 && ! hayGanador ) {
        hayGanador = true;
        jlblGanador.setVisible ( true );
    }
}
// -----
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for ( javax.swing.UIManager.LookAndFeelInfo info :
              javax.swing.UIManager.getInstalledLookAndFeels() ) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(HiloFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(HiloFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(HiloFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```
java.util.logging.Logger.getLogger(HiloFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new HiloFrame().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JButton jButtonIniciarHilo;
private javax.swing.JLabel jLabelGanador1;
private javax.swing.JLabel jLabelGanador2;
private javax.swing.JLabel jLabelGanador3;
private javax.swing.JProgressBar jpbrHilo1;
private javax.swing.JProgressBar jpbrHilo2;
private javax.swing.JProgressBar jpbrHilo3;
private javax.swing.JTextField jtxfHilo1;
private javax.swing.JTextField jtxfHilo2;
private javax.swing.JTextField jtxfHilo3;
// End of variables declaration
}
```

EJERCICIO 2

Pelota.java

```
/*-----
*:
*:      TECNOLOGICO NACIONAL DE MEXICO
*:      INSTITUTO TECNOLOGICO DE LA LAGUNA
*:      INGENIERIA EN SISTEMAS COMPUTACIONALES
*:      TOPICOS AVANZADOS DE PROGRAMACION "B"
*:
*:      SEMESTRE: AGO-DIC/2020   HORA: 17-18 HRS
*:
*:      Clase Pelota que permite dibujar una pelota en un Frame.
*:
*: Archivo      : Pelota.java
*: Autor       : Félix Gerardo Martínez Hinojo   17130800
*: Fecha       : 13/ENE/2021
```

```

:* Compilador : JAVA J2SE v1.8.0
:* Descripción : Clase Java con la parte lógica programada de la Pelota creada mediante Hilos.

```

```

:*

```

```

:* Última modif:

```

```

:* Fecha      Modificación      Motivo

```

```

:* =====

```

```

=====

```

```

:* 13/ENE/2021 Félix Mtz      Se añadió el prologo.

```

```

:* -----*/

```

```

package pelotas;

```

```

import java.awt.*;

```

```

import java.util.logging.Level;

```

```

import java.util.logging.Logger;

```

```

import javax.swing.JPanel;

```

```

// -----

```

```

public class Pelota extends Thread {

```

```

    private JPanel      jpnMarcoPelota;

```

```

    int                x;

```

```

    int                y;

```

```

    int                ancho, alto;

```

```

    int                despX, despY;

```

```

    int                periodo;

```

```

    boolean            seguirRebotando = true;

```

```

    private Graphics    lienzo;

```

```

// -----

```

```

// Constructor de la clase Pelota que recibe el contexto del Panel.

```

```

public Pelota ( JPanel jpnMarcoPelota ) {

```

```

    super ();

```

```

    this.jpnMarcoPelota = jpnMarcoPelota;

```

```

    lienzo = jpnMarcoPelota.getGraphics ();

```

```

    ancho = jpnMarcoPelota.getWidth ();

```

```

    alto = jpnMarcoPelota.getHeight ();

```

```

    x = ancho / 2;

```

```

    y = alto / 2;

```

```

    despX = 1 + ( int ) ( Math.random () * 7 );

```

```

    despY = 1 + ( int ) ( Math.random () * 3 );

```

```

    periodo = 10 + ( int ) ( Math.random () * 40 );

```

```

}

```

```

// -----

```

```

// Método run () que pone en marcha los hilos para las pelotas.

```

```

@Override

```

```

public void run () {

```

```
while ( seguirRebotando ) {
    mover ();
    rebotar ();
    limpiar ();
    dibujar ();
    pausa ();
}
}
// -----
// Método mover () que mueve las pelotas de una posición a otra.
private void mover () {
    x += despX;
    y += despY;
}
// -----
// Método rebotar () que rebota las pelotas al llegar a las posiciones 0,0 del panel.
private void rebotar () {
    if ( x <= 0 || x >= ancho ) {
        despX *= -1;
    }

    if ( y <= 0 || y >= alto ) {
        despY *= -1;
    }
}
// -----
// Método limpiar () que rellena espacio rectangular en blanco para limpiar el panel de tal forma
// que el rastro de la pelota sea eliminado del panel.
private void limpiar () {
    lienzo.setColor ( Color.white );
    lienzo.fillRect ( 0, 0, ancho, alto );
}
// -----
// Método dibujar () que se encarga de dibujar o pintar la pelota dentro del panel.
private void dibujar () {
    lienzo.setColor ( Color.red );
    lienzo.fillOval ( x, y, 20, 20 );
}
// -----
// Método pausa () que pone pausa al movimiento de la pelota dibujada en el panel.
private void pausa () {
    try {
        Thread.sleep ( periodo );
    } catch (InterruptedException ex) {
        Logger.getLogger ( Pelota.class.getName () ).log ( Level.SEVERE, null, ex );
    }
}
```

}

PelotaRebotandoBean.java

```

/*-----
*:          TECNOLOGICO NACIONAL DE MEXICO
*:          INSTITUTO TECNOLOGICO DE LA LAGUNA
*:          INGENIERIA EN SISTEMAS COMPUTACIONALES
*:          TOPICOS AVANZADOS DE PROGRAMACION "B"
*:
*:          SEMESTRE: AGO-DIC/2020   HORA: 17-18 HRS
*:
*:          Bean que permite dibujar una pelota en un Frame.
*:
*: Archivo   : PelotaRebotandoBean.java
*: Autor     : Félix Gerardo Martínez Hinojo   17130800
*: Fecha     : 13/ENE/2021
*: Compilador : JAVA J2SE v1.8.0
*: Descripción : Panel con la parte híbrida ( visual / lógica ) programada de las pelotas rebotando
*:              creadas mediante Hilos.
*:
*: Última modificación:
*: Fecha      Modificación      Motivo
*:=====
*: 13/ENE/2021 Félix Mtz      Se añadió el prologo.
*:-----*/
package pelotas;

// -----
public class PelotaRebotandoBean extends javax.swing.JPanel {

    private Pelota pelota;

    public PelotaRebotandoBean() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jpnMarcoPelota = new javax.swing.JPanel();
        jbtnIniciar = new javax.swing.JButton();
        jbtnDetener = new javax.swing.JButton();

        jpnMarcoPelota.setBackground(new java.awt.Color(255, 255, 255));

```



```
jpnMarcoPelota.setBorder(javax.swing.BorderFactory.createEtchedBorder());

javax.swing.GroupLayout jpnMarcoPelotaLayout = new javax.swing.GroupLayout(jpnMarcoPelota);
jpnMarcoPelota.setLayout(jpnMarcoPelotaLayout);
jpnMarcoPelotaLayout.setHorizontalGroup(
    jpnMarcoPelotaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(
            javax.swing.GroupLayout.Alignment.LEADING,
            .addGroup(
                javax.swing.GroupLayout.Alignment.LEADING,
                .addGap(0, 0, Short.MAX_VALUE)
            )
        )
);
jpnMarcoPelotaLayout.setVerticalGroup(
    jpnMarcoPelotaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(
            javax.swing.GroupLayout.Alignment.LEADING,
            .addGap(0, 248, Short.MAX_VALUE)
        )
);

jbtnIniciar.setText("Iniciar");
jbtnIniciar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jbtnIniciarActionPerformed(evt);
    }
});

jbtnDetener.setText("Detener");
jbtnDetener.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jbtnDetenerActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);
this.setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(
            layout.createSequentialGroup()
                .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .add(
                        layout.createSequentialGroup()
                            .add(jpnMarcoPelota, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addContainerGap()
                        )
                    .add(
                        layout.createSequentialGroup()
                            .add(jbtnIniciar, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addContainerGap()
                        )
                )
                .add(jbtnDetener, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGap(29, 29, 29)
        )
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .add(
            layout.createSequentialGroup()
                .add(jpnMarcoPelota, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .add(jbtnIniciar, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .add(jbtnDetener, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGap(29, 29, 29)
            )
        )
);
```

```

        .addComponent(jpnMarcoPelota,
                        javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jbtnIniciar)
            .addComponent(jbtnDetener))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
} // </editor-fold>
// -----
// Evento click del botón iniciar de el frame.
private void jbtnIniciarActionPerformed(java.awt.event.ActionEvent evt) {
    pelota = new Pelota ( jpnMarcoPelota );
    pelota.start ();
}
// -----
// Evento click del botón detener del frame.
private void jbtnDetenerActionPerformed(java.awt.event.ActionEvent evt) {
    pelota.seguirRebotando = false;
}

// Variables declaration - do not modify
private javax.swing.JButton jbtnDetener;
private javax.swing.JButton jbtnIniciar;
private javax.swing.JPanel jpnMarcoPelota;
// End of variables declaration
}

```

PelotasRebotandoFrame.java

```

/*-----
*:
*:          TECNOLOGICO NACIONAL DE MEXICO
*:          INSTITUTO TECNOLOGICO DE LA LAGUNA
*:          INGENIERIA EN SISTEMAS COMPUTACIONALES
*:          TOPICOS AVANZADOS DE PROGRAMACION "B"
*:
*:          SEMESTRE: AGO-DIC/2020   HORA: 17-18 HRS
*:
*:          Formulario visual que visualiza una pelota dibujada mediante su clase lógica
*:          Pelota.
*:
*: Archivo   : PelotaRebotandoFrame.java
*: Autor     : Félix Gerardo Martínez Hinojo   17130800
*: Fecha     : 13/ENE/2021
*: Compilador : JAVA J2SE v1.8.0
*: Descripción : Formulario con la parte visual programada de las pelotas dibujadas mediante Hilos.

```



```

        .addComponent(pelotaRebotandoBean2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE)))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(pelotaRebotandoBean2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(pelotaRebotandoBean1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(pelotaRebotandoBean3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addContainerGap())
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                    .addGap(0, 0, Short.MAX_VALUE)
                    .addComponent(pelotaRebotandoBean4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))))
    );

    pack();
} // </editor-fold>

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(PelotasRebotandoFrame.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
    } catch (InstantiationException ex) {

```

```

java.util.logging.Logger.getLogger(PelotasRebotandoFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(PelotasRebotandoFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(PelotasRebotandoFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new PelotasRebotandoFrame().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private pelotas.PelotaRebotandoBean pelotaRebotandoBean1;
private pelotas.PelotaRebotandoBean pelotaRebotandoBean2;
private pelotas.PelotaRebotandoBean pelotaRebotandoBean3;
private pelotas.PelotaRebotandoBean pelotaRebotandoBean4;
// End of variables declaration
}

```

EJERCICIO 3

Globo.java

```

/*-----
*:          TECNOLÓGICO NACIONAL DE MÉXICO
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: AGO-DIC/2020   HORA: 17-18 HRS
*:
*:          Clase Globo que permite dibujar un globo en un Frame y/o Panel.
*:
*: Archivo   : Globo.java
*: Autor     : Félix Gerardo Martínez Hinojo   17130800
*: Fecha     : 14/ENE/2021
*: Compilador : JAVA J2SE v1.8.0
*: Descripción : Clase Globo con la parte lógica programada de los globos creados mediante Hilos.

```

```

:*
:* Última modif:
:* Fecha      Modific  Motivo
:*=====
==
:* 14/ENE/2021 Félix Mtz      Se añadió el prologo.
:*-----*/
package globos;

import java.awt.*;
import java.util.logging.Level;
import java.util.logging.Logger;

// -----
public class Globo extends Thread {
    GlobosFrame parent;
    Color    color;
    int      radioInicial;
    int      radioActual;
    int      periodoSleep;
    int      x;
    int      y;
    boolean   creciendo;

    // Constructor de la clase Globo para dibujarlo en el frame.
    public Globo ( GlobosFrame parent, Color color, int radioInicial, int periodoSleep, int x , int y ) {
        this.parent    = parent;
        this.color      = color;
        this.radioInicial = radioInicial;
        this.radioActual = radioInicial;
        this.periodoSleep = periodoSleep;
        this.x          = x;
        this.y          = y;
        this.creciendo   = ( Math.random() > 0.5 ) ? true : false;
    }

    // -----
    // Método run () que poner a ejecutar el Hilo.
    @Override
    public void run () {
        while ( true ) {
            if ( creciendo )
                radioActual++;
            else
                radioActual--;

            if ( radioActual < 5 || radioActual >= radioInicial * 2 )
                creciendo = !creciendo;

            parent.repaint ();

            try {
                Thread.sleep ( periodoSleep );
            } catch ( InterruptedException ex ) {
                System.out.println ( ex.getMessage () );
            }
        }
    }
}

```

```

    }
}
// -----
// Método dibujar () que realiza el pintado del globo en el frame.
public void dibujar ( Graphics g ) {
    g.setColor ( color );
    g.fillOval ( x - radioActual, y - radioActual, 2 * radioActual, 2 * radioActual );
}
}

```

GlobosFrame.java

```

/* -----
*:          TECNOLOGICO NACIONAL DE MEXICO
*:          INSTITUTO TECNOLOGICO DE LA LAGUNA
*:          INGENIERIA EN SISTEMAS COMPUTACIONALES
*:          TOPICOS AVANZADOS DE PROGRAMACION "B"
*:
*:          SEMESTRE: AGO-DIC/2020   HORA: 17-18 HRS
*:
*:          Formulario que dibuja globos visuales.
*:
*: Archivo   : GlobosFrame.java
*: Autor     : Félix Gerardo Martínez Hinojo   17130800
*: Fecha     : 14/ENE/2021
*: Compilador : JAVA J2SE v1.8.0
*: Descripción : Clase Frame con la parte visual programada de los globos creados mediante Hilos, utiliza la clase lógica
Globo.
*:
*: Última modif:
*: Fecha      Modificó      Motivo
*: =====
==
*: 14/ENE/2021 Félix Mtz     Se añadió el prologo.
*: -----*/
package globos;

import java.awt.*;

// -----
public class GlobosFrame extends javax.swing.JFrame {

    public static final int MAX_GLOBOS = 10;
    private Globo [] globos    = new Globo [ MAX_GLOBOS ];
    private Color [] colores   = new Color [] {
        Color.BLUE, Color.CYAN, Color.GRAY, Color.GREEN, Color.MAGENTA, Color.ORANGE, Color.PINK,
        Color.RED, Color.YELLOW
    };

    private int  ancho;
    private int  alto;
    private Image image;
    private Graphics grImage;

    public GlobosFrame() {
        initComponents();
    }
}

```

```

ancho = this.getWidth ();
alto = this.getHeight ();

for ( int i = 0; i < MAX_GLOBOS; i++ ) {
    Color color = colores [ ( int ) ( Math.random() * 8 ) ];
    int radio = 5 + ( int ) ( Math.random() * alto * 0.2 );
    int sleep = 10 + ( int ) ( Math.random() * 40 );
    int x = radio + ( int ) ( Math.random() * ( ancho - radio * 2 ) );
    int y = radio + ( int ) ( Math.random() * ( alto - radio * 2 ) );

    globos [ i ] = new Globo ( this, color, radio, sleep, x, y );
    globos [ i ].start ();
}
}
// -----
@Override
public void paint( Graphics g ) {
//    super.paint ( g );

    if ( image == null ) {
        image = this.createImage ( ancho, alto );
        grpImage = image.getGraphics ();
    }

    grpImage.setColor( Color.WHITE );
    grpImage.fillRect ( 0, 0, ancho, alto );

    for ( int i = 0; i < MAX_GLOBOS; i++ ) {
        globos [ i ].dibujar ( grpImage );
    }

    g.drawImage ( image, 0, 0, null );
}
// -----

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Globos");

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(0, 897, Short.MAX_VALUE)
            )
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(0, 535, Short.MAX_VALUE)
            )
    );
}

```



```

    pack();
} // </editor-fold>

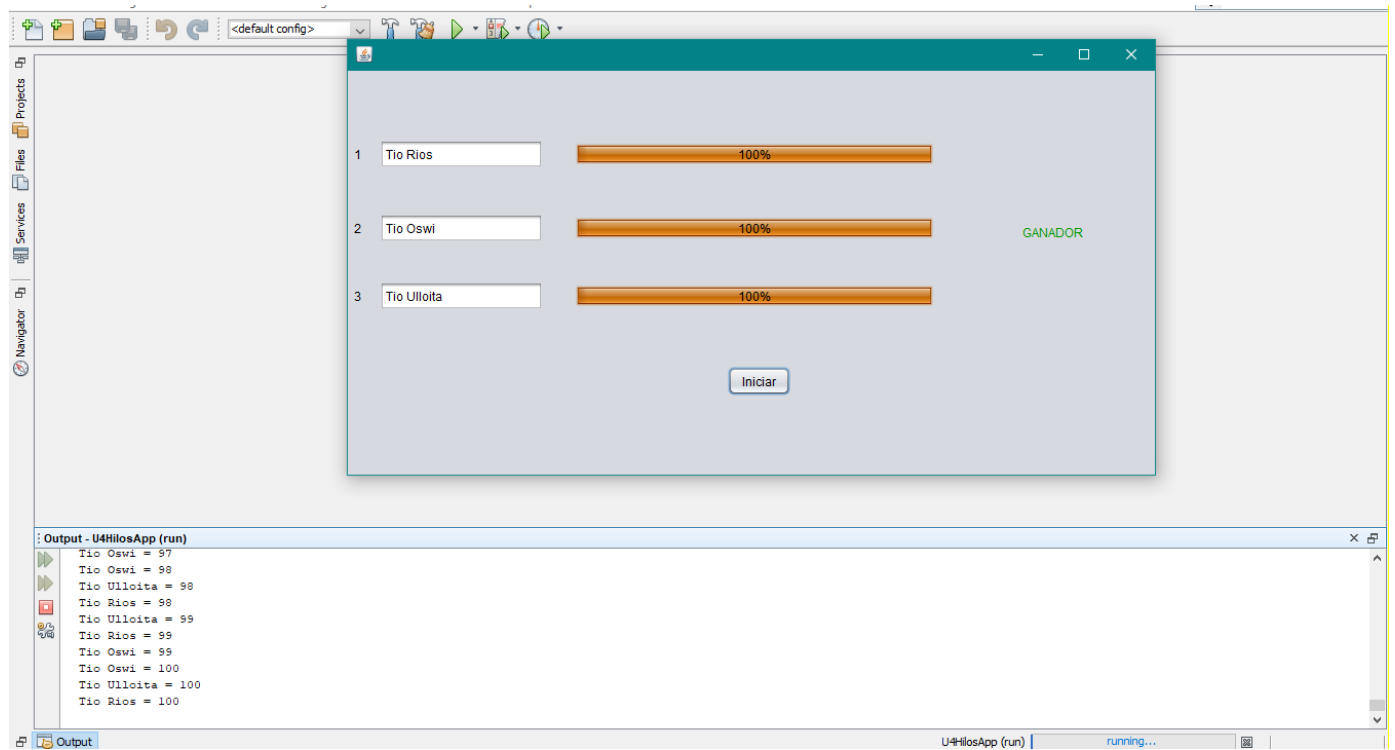
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(GlobosFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(GlobosFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(GlobosFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(GlobosFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
} //</editor-fold>

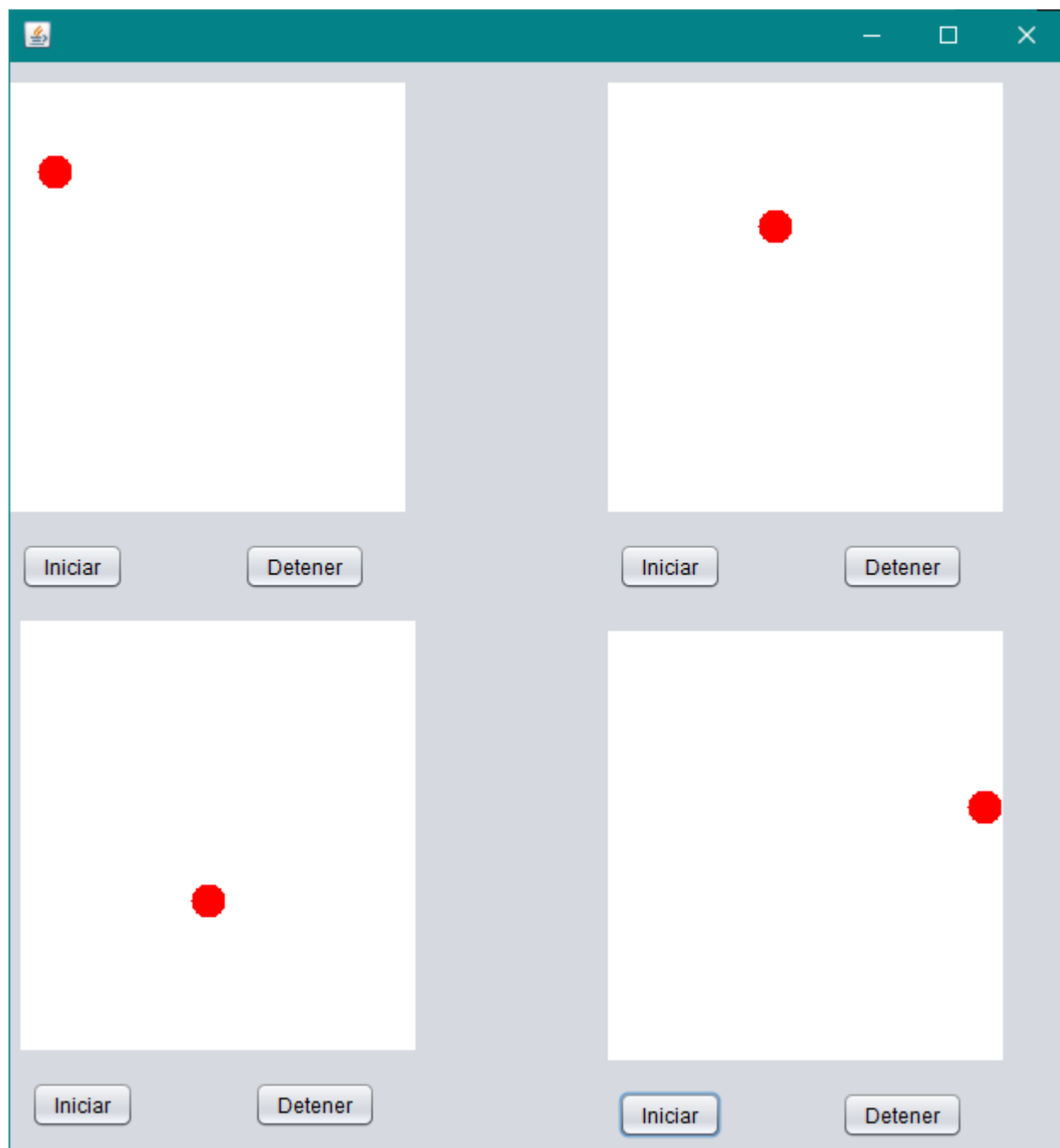
/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new GlobosFrame().setVisible(true);
    }
});
}

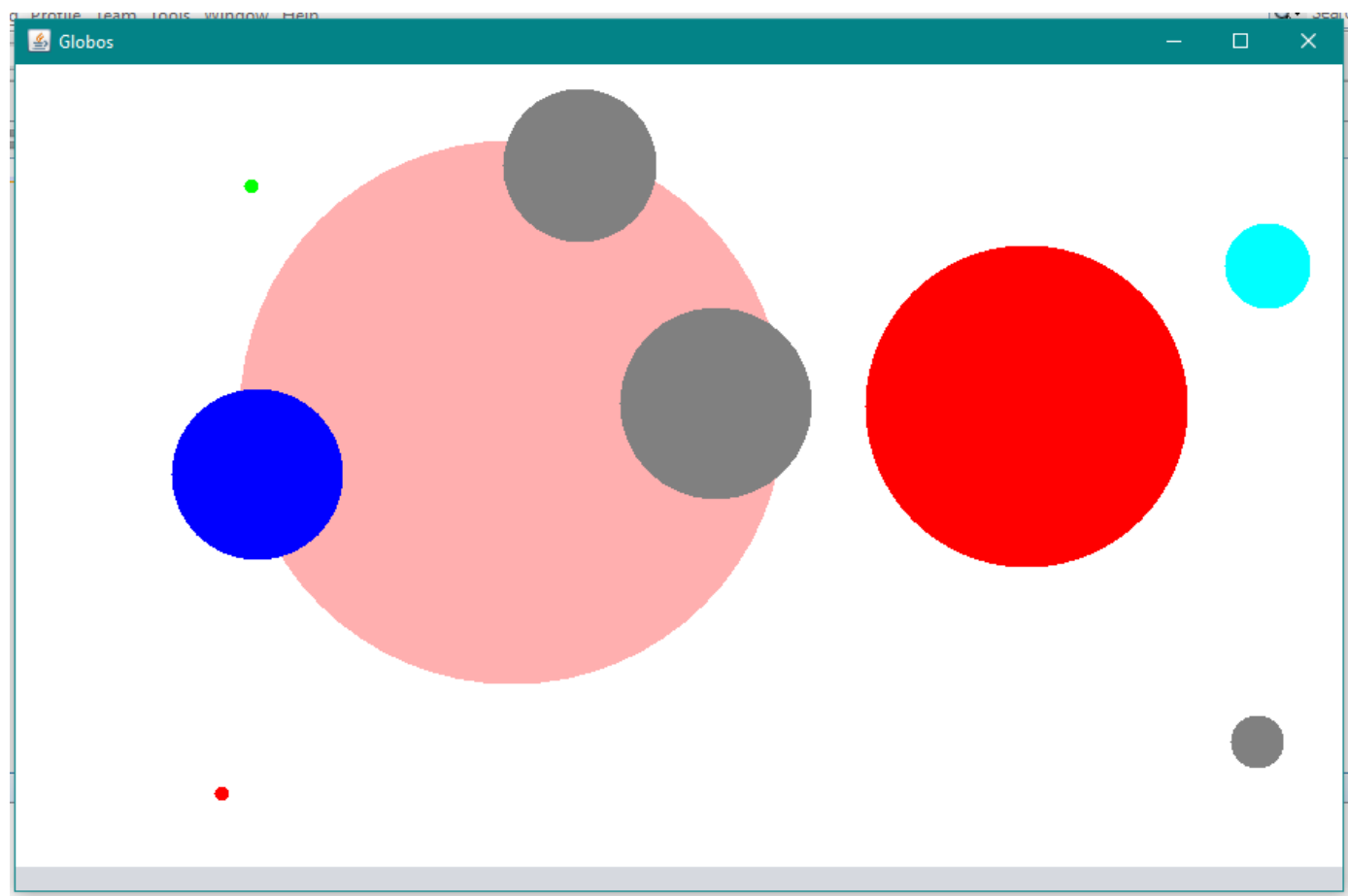
// Variables declaration - do not modify
// End of variables declaration
}

```

Prueba de Ejecución







-oOo-