Diplomado en Java

Semana 4

Erandi Castillo Montiel

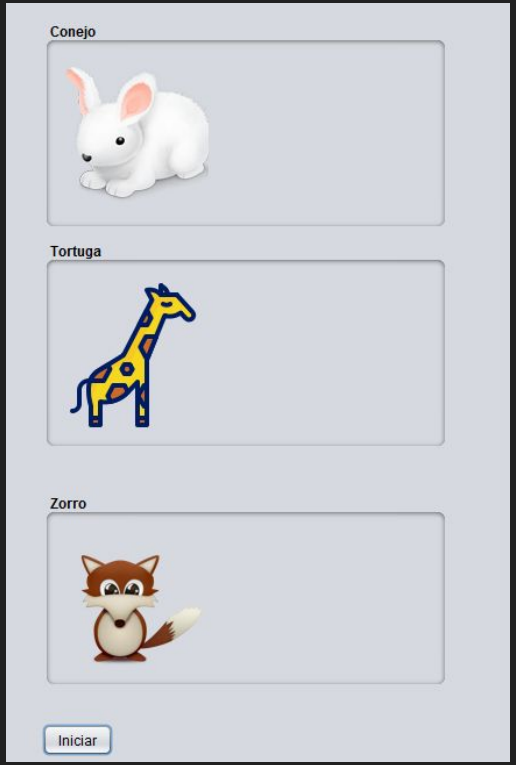
Carrera de animales - thread

```java
public class AnimalThread extends Thread {

    private String nombre;
    private int limite;

    public AnimalThread(String nombre, int limite) {
        this.nombre = nombre;
        this.limite = limite;
    }

    @Override
    public void run() {
        for (int i = 0; i < limite; i++) {
                System.out.println(nombre + " avanza");
        }
        System.out.println(nombre + " ha llegado a la meta ");
        yield(); // alterna el procesamiento de los hilos
    }
}
```

Main

```
public class Animales {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        AnimalThread conejo = new AnimalThread("conejo",100);
        AnimalThread tortuga = new AnimalThread("tortuga",100);
        AnimalThread zorro = new AnimalThread("zorro",100);

        conejo.start();
        tortuga.start();
        zorro.start();
        System.out.println("Se ha terminado la carrera");
    }

}
```

Carrera de animales - GUI Thread



AnimalAThread - clase

```java
public class AnimalAThread extends Thread {
private String nombre;
private int limite;
private JLabel label;
private int retardo;

public AnimalAThread(String nombre, int limite, JLabel label, int retardo){
    this.nombre=nombre;
    this.limite=limite;
    this.label=label;
    this.retardo=retardo;
}
```

run()

```java
@Override
public void run(){
    for (int i = 0; i < limite; i++) {
        try {
            System.out.println(nombre + " avanza");
            label.setLocation(i, 0);
            Thread.sleep(retardo);
        } catch (InterruptedException ex) {
            Logger.getLogger(AnimalAThread.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    System.out.println(nombre + " ha llegado a la meta ");
    JOptionPane.showMessageDialog(null, nombre + " ha llegado a la meta ");
    yield(); // alterna el procesamiento de los hilos
}
```

Interfaz gráfica

Inicializa la interfaz

3 JPanel , 3 JLabel y 1 botón

```
 */
public Carrera() {
    initComponents();

    ImageIcon icon = new ImageIcon (getClass().getResource("/imagenes/conejito.png"));
    ImageIcon tamaño = new ImageIcon(icon.getImage().getScaledInstance(5, 5, 1));
//  ImageIcon icon = new ImageIcon(new ImageIcon("/imagenes/conejito.png").getImage().getScaledInstance(35, 35, Image.SCALE_REPLICATE));
    labelConejo.setIcon(icon);
     ImageIcon icon2 = new ImageIcon (getClass().getResource("/imagenes/jirafa.png"));
    ImageIcon tamaño2 = new ImageIcon(icon2.getImage().getScaledInstance(5, 5, 1));
     labelTortuga.setIcon(icon2);

     ImageIcon icon3 = new ImageIcon (getClass().getResource("/imagenes/zorrito.png"));
    ImageIcon tamaño3 = new ImageIcon(icon3.getImage().getScaledInstance(5, 5, 1));
    labelZorro.setIcon(icon3);
}
```

```java
private void btnIniciarActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO code application logic here
    AnimalAThread conejo = new AnimalAThread("conejo",100,labelConejo,100);
    AnimalAThread tortuga = new AnimalAThread("tortuga",100,labelTortuga,30);
    AnimalAThread zorro = new AnimalAThread("zorro",100,labelZorro,60);

    conejo.start();
    tortuga.start();
    zorro.start();
}
```

ActionPerformed - botón Iniciar

Main

```java
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
     Look and feel setting code (optional)

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

Animación - pelota

JPanel - Cambiar fondo blanco

```java
public class LienzoPelota extends javax.swing.JPanel implements Runnable {

    /**
     * Creates new form LienzoPelota
     */
    Thread hilo;
    //se mueve de forma horizontal
    int x = getWidth()/ 2; //inicializada a la mitad de lienzo
    public LienzoPelota() {
        initComponents();
        hilo = new Thread(this);
    }

    public void paint (Graphics g){
        //cuando se inicie la animación no se vuelva a repetir la imagen en el lienzo
        g.setColor(getBackground()); //para que se vuelva a pintar la imagen
        g.fillRect(0,0,getWidth(),getHeight()); //ovalo con el ancho y algo del panel

        g.setColor(Color.red);//color de la pelota
        g.fillOval(x, getHeight()/2,30, 30);  //dibuja la pelota en la posición X a la mitad del lienzo
                                              //altura y ancho constante 30
    }
```

Acciones- run()

```java
public void inicio (){
    hilo.start();

}

public void pausa(){
    hilo.suspend();
}

public void continuar(){
    hilo.resume();
}
```

```java
@Override
public void run() {  //animación
        try {
            while(true ){
                while(x<getWidth()-30){ //mueve a la derecha
                    Thread.sleep(50);  //contrala la velocidad de la animación
                    x+=10;              //se mueve en escala de 10
                    repaint();         //repinte
                }
                while(x>10){             //mueva a la izquierda,impresión uqe toca la pared
                    Thread.sleep(50);
                    x-=10;
                    repaint();
                }

            }
    } catch (Exception e) {
            System.out.println("Sucedion un error"+e.getMessage());
    }

}
```

Botones del GUI

```
private void btnInicioActionPerformed(java.awt.event.ActionEvent evt) {
  lienzoPelota1.inicio();
}

private void btnContinuarActionPerformed(java.awt.event.ActionEvent evt) {
   lienzoPelota1.continuar();
}

private void btnPausarActionPerformed(java.awt.event.ActionEvent evt) {
    lienzoPelota1.pausa();
}
```

Animación pelotas 2

JPanel - fondo blanco

```
public class LienzoPelota2 extends javax.swing.JPanel implements Runnable {
    int x=0,y=0, velX=2,velY=2;

     Thread hilo,hilo2;

    public void paint(Graphics g ){
        g.setColor(getBackground());
        g.fillRect(0,0,getWidth(),getHeight());
        System.out.println("width "+getWidth() + " height "+getHeight());
        g.setColor(Color.BLUE);
        g.fillOval(x, y,40, 40);
    }
```

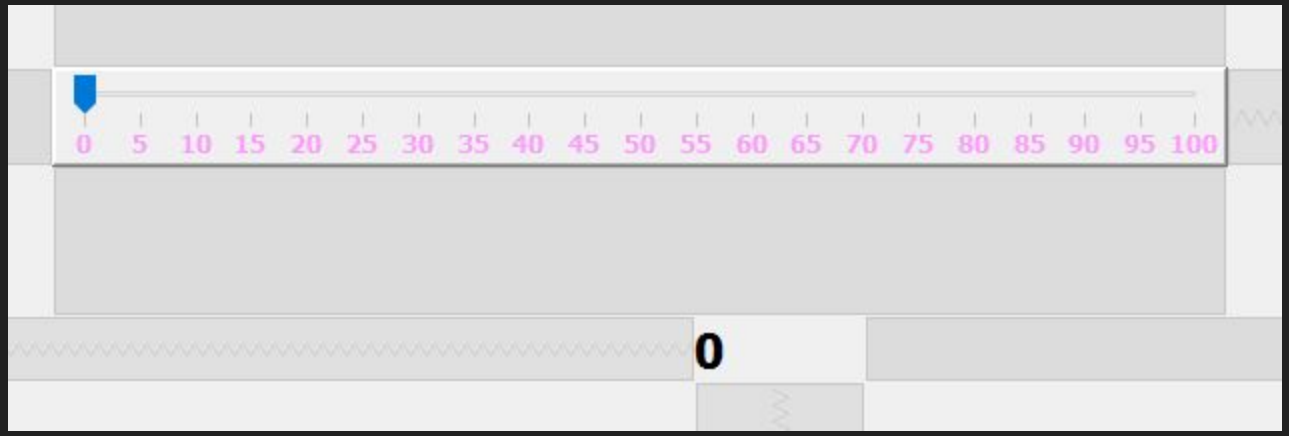
Acciones

```
    public LienzoPelota2() {
        initComponents();
         hilo = new Thread(this);

    }
public void inicio (){
        hilo.start();
    }

    public void pausa(){
        hilo.suspend();
      }

    public void continuar(){
        hilo.resume();
    }
```

Run ()

```
@Override
public void run() {
   try {
           while(true ){

               if(x<(getWidth()-40) || y<(getHeight()-40)){
                Thread.sleep(50);
                    x+=velX;              //se mueve en escala de 10
                    y+=velY;
                    System.out.println("!! x: "+x  +"y: "+y );

                    repaint();
               }
                if(x<=0||x>=getWidth()-40){

                velX= -velX;
                System.out.println("--");
                }
                if(y<=0||y>=getHeight()-40){
                   velY= -velY;
                   System.out.println("**");
               }


           }
    } catch (Exception e) {
           System.out.println("Sucedion un error"+e.getMessage());
    }
}
```

Slider



stateChanged

```
private void jSlider2StateChanged(javax.swing.event.ChangeEvent evt) {

    labelSlider.setText(Integer.toString(jSlider2.getValue()));
}
```

Slider 2


R=0
0 20 40 60 80 100 120 140 160 180 200 220 240
G=0
0 20 40 60 80 100 120 140 160 180 200 220 240
B=0
0 20 40 60 80 100 120 140 160 180 200 220 240

Properties

Properties	
background	[240,240,240]
font	Tahoma 11 Plain
foreground	[204,0,0]
majorTickSpacing	20
maximum	255
minimum	0
minorTickSpacing	0
orientation	HORIZONTAL
paintLabels	☑
paintTicks	☑
paintTrack	☑
snapToTicks	☐
toolTipText	

Jlabel y cambiaColor()

```
private void jSlider4StateChanged(javax.swing.event.ChangeEvent evt) {
    cambiarColor();
    labelR.setText(Integer.toString(jSlider4.getValue()));
}

private void jSlider1StateChanged(javax.swing.event.ChangeEvent evt) {
    cambiarColor();
    labelG.setText(Integer.toString(jSlider1.getValue()));
}

private void jSlider2StateChanged(javax.swing.event.ChangeEvent evt) {
    cambiarColor();
    labelB.setText(Integer.toString(jSlider2.getValue()));
}
```

cambiaColor()

```
public void cambiarColor(){
 int R,G,B;
    R=jSlider4.getValue();
    G=jSlider1.getValue();
    B=jSlider2.getValue();

    jTextField1.setBackground(new Color(R,G,B));

}
```

SQLite

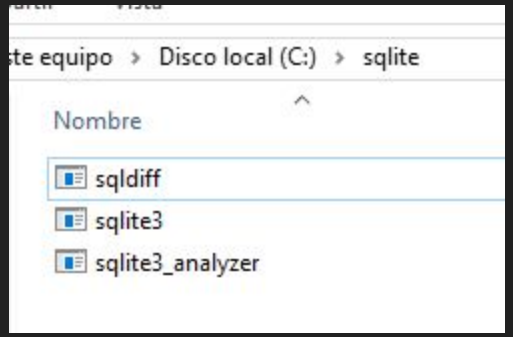
Obtener el binario https://www.sqlite.org/download.html

Precompiled Binaries for Windows

sqlite-dll-win32-x86-3240000.zip (444.18 KiB)	32-bit DLL (x86) for SQLite version 3.24.0. (sha1: 11d37a0eccbaf9995c6b236ff1a99d174a2566bd)
sqlite-dll-win64-x64-3240000.zip (736.78 KiB)	64-bit DLL (x64) for SQLite version 3.24.0. (sha1: 534776011cb664a03009bca76c06c9855c1d15a9)
sqlite-tools-win32-x86-3240000.zip (1.64 MiB)	A bundle of command-line tools for managing SQLite databas (sha1: e87b1642d9b36fcbe39b003524e58a22c1fe4b54)

Crear carpeta sqlite

Descomprimir zip



Desde la terminal

```
C:\sqlite>sqlite3
SQLite version 3.24.0 2018-06-04 19:24:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

Crear carpeta sqlite

Descomprimir

Abrir en la terminal

Ejecutar comando sqlite3

Instalar DB browser for Sqlite

https://sqlitebrowser.org/

Crear una base de datos pruebaDB

Con una tabla Alumno,

Propiedades Nombre, Apellido, Edad, Calificacion, EdoCivil

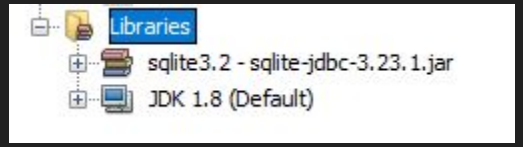
Name	Type	Schema
Tables (1)		
Alumno		CREATE TABLE `Alumno` (`Nom
Nombre	TEXT	`Nombre` TEXT
Apellido	TEXT	`Apellido` TEXT
EdoCivil	TEXT	`EdoCivil` TEXT
Edad	INTEGER	`Edad` INTEGER
Promedio	REAL	`Promedio` REAL

Descargar y agregar la librería al proyecto

https://bitbucket.org/xerial/sqlite-jdbc/downloads/

http://www.sqlitetutorial.net/download-install-sqlite/

https://bitbucket.org/xerial/sqlite-jdbc



Clase conector - conector a la BD

Constructor y propiedades

```java
public class Conector {
    // String url="/Users/erandi/GitHub/SWING/SQLite/pruebaBD.db";
    String url = "C:\\Users\\erand\\Documents\\CursoDiplomanoJava_SWING\\SWING\\SQLite\\pruebaBD.db";
    Connection connect;

    public void connect() {
        try {
            connect = DriverManager.getConnection("jdbc:sqlite:" + url);
            if (connect != null) {
                System.out.println("Conectado");
            }
        } catch (SQLException ex) {
            System.err.println("No se ha podido conectar a la base de datos\n" + ex.getMessage());
        }
    }
```

Mostrar alumno

```java
public void mostrarAlumnos() {
    ResultSet result = null;
    try {
        PreparedStatement st = connect.prepareStatement("select * from Alumno");
        result = st.executeQuery();
        while (result.next()) {

            System.out.print("Nombre: ");
            System.out.println(result.getString("Nombre"));

            System.out.print("Apellidos: ");
            System.out.println(result.getString("Apellido"));

            System.out.println("========================");
        }
    } catch (SQLException ex) {
        System.err.println(ex.getMessage());
    }
}
```

Salvar alumno - execute()

```
public void saveAlumno(Alumno alumno) {
    try {
        PreparedStatement st = connect.prepareStatement("insert into Alumno "
                + "(Nombre,Apellido,EdoCivil,Promedio,Edad) values (?,?,?,?,?)");
        st.setString(1, alumno.getNombre());
        st.setString(2, alumno.getApellido());
        st.setString(3, alumno.getEdoCivil());
        st.setDouble(4, alumno.getPromedio());
        st.setInt(5, alumno.getEdad());
        st.execute();
    } catch (SQLException ex) {
        System.err.println(ex.getMessage());
    }

}
```

Cerrar conexión

```
public void close() {
    try {
        connect.close();
    } catch (SQLException ex) {
        Logger.getLogger(Conector.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Clase alumno

```java
public class Alumno {
    private String nombre;
    private String apellido;
    private int edad;
    private String edoCivil;
    private double promedio;

    public Alumno(String nombre, String apellido, int edad, String edoCivil, double promedio) {
        this.nombre = nombre;
        this.apellido = apellido;
        this.edad = edad;
        this.edoCivil = edoCivil;
        this.promedio = promedio;
    }
```

Métodos adicionales

```
public void save(){
        Conector con = new Conector();
        con.connect();
        con.saveAlumno(this);
        con.close();
    }

 public void consultDatos(){
        Conector con = new Conector();
        con.connect();
        con.mostrarAlumnos();
        con.close();
    }
```

Main

```
public class TestSQLite {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Alumno alumno = new Alumno("Atxy2k", "SerProgramador.es", 2, "hi", 2.4);
        alumno.save();
        alumno.consultDatos();

    }

}
```