



Universidad de las Fuerzas Armadas (ESPE)

Programación Orientada a Objetos

Actividad Experimental N.º 1 Primer Parcial

Tema: Sistema de Gestión de Proyectos de TIC

Autores:

Israel Fernando Portilla Santamaria

Erick Sebastián Ramírez Tasiguano

Jair Paul Loor Ramirez

Juan Fabricio Pilataxi Tipan

Carlos Danilo Collaguazo Pinza

domingo, 8 de diciembre de 2024

Actividad-Sistema de Gestión de Proyectos

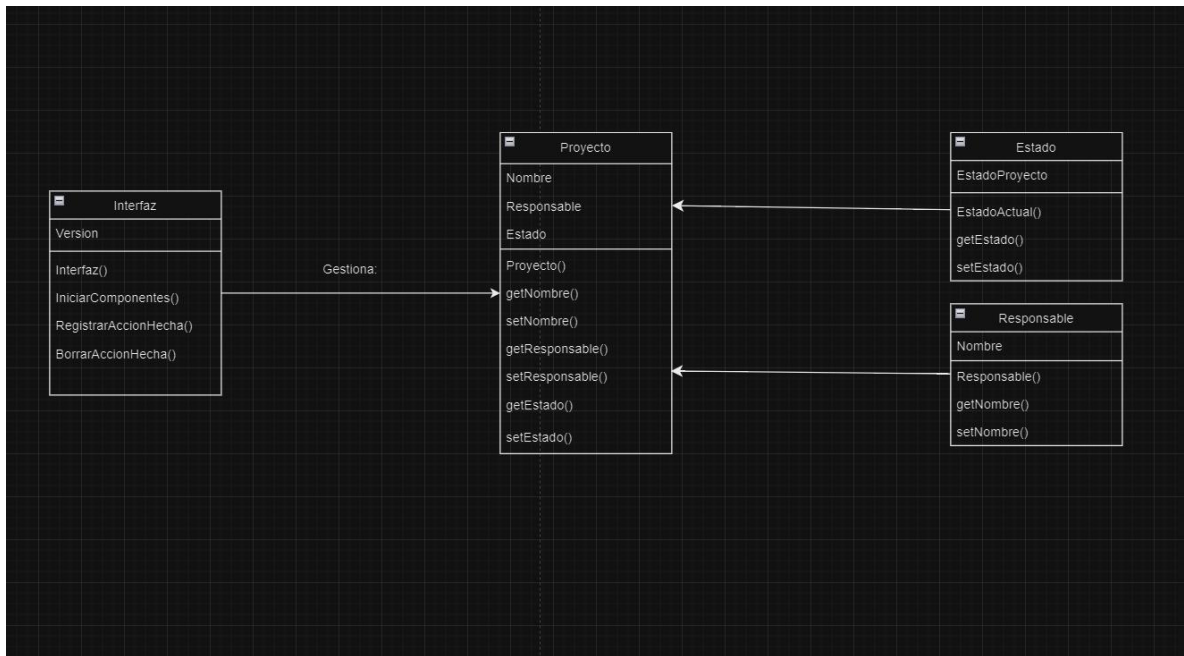
Objetivo: Implementar un sistema de gestión de proyectos utilizando el paradigma de programación orientado a objetos. Incluya las funcionalidades de registrar proyectos, asignar tareas y mostrar su estado de forma organizada

Introducción: El paradigma orientado a objetos (POO) de forma resumida dice que todo es un objeto y que la composición básica del código es clases e instancias de estas(objetos), es decir, el registro de proyectos, asignación de tareas y el mostrar el estado de los proyectos los debemos tratar como objetos de los cuales como se a desarrollado en las clases con el docente tenemos que asignar atributos y métodos que nos permitan darles vida y que sean funcionales. En este informe detallaremos como implementamos el paradigma POO con lo solicitado, utilizaremos algunas herramientas que nos facilitaron la codificación y la creación del UML, en este caso son Neatbeans y Draw.io. Por último, explicaremos un poco sobre los sistemas de gestión de proyectos, es una actividad fundamental en diversos ámbitos, ya que permite planificar, organizar y controlar las tareas necesarias para alcanzar objetivos específicos, es clave para garantizar el éxito de proyectos en diversos campos, desde tecnología y construcción hasta educación y servicios.

Desarrollo:

Como ya se mencionó en la introducción el pilar fundamental para esta actividad será el paradigma POO utilizaremos sus conceptos y definiciones para la actividad. El desarrollo del sistema se realizó en varias etapas, garantizando una estructura organizada. A continuación, cada una de estas etapas:

1. Creación del UML



2. Código

```
package gestion_proyectos;

//Importo javax.swing que voy a usar para la tabla y modelos
import java.awt.HeadlessException;
import javax.swing.JOptionPane;
import static javax.swing.JOptionPane.showMessageDialog;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author jair
 */
public class Interfaz extends javax.swing.JFrame {

    /**
     * Creates new form Interfaz
     */
    public Interfaz() {
        initComponents();
    }
}
```

```

        model=(DefaultTableModel)this.TableProyecto.getModel();
    }
    DefaultTableModel model;
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jPanel2 = new javax.swing.JPanel();
        btnRegistrar = new javax.swing.JButton();
        btnBorrar = new javax.swing.JButton();
        txtNombre = new javax.swing.JTextField();
        txtResponsable = new javax.swing.JTextField();
        txtEstado = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jScrollPane2 = new javax.swing.JScrollPane();
        TableProyecto = new javax.swing.JTable();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jPanel1.setBackground(new java.awt.Color(153, 153, 153));
        jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
        getContentPane().add(jPanel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(790, 0, -1, 388));

        jPanel2.setBackground(new java.awt.Color(153, 153, 153));

```

```

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(0, 800, Short.MAX_VALUE)
        .addContainerGap())
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(0, 70, Short.MAX_VALUE)
            .addContainerGap())
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(0, 800, 70))
            .addContainerGap()));

        btnRegistrar.setBackground(new java.awt.Color(0, 102, 102));
        btnRegistrar.setForeground(new java.awt.Color(204, 204, 204));
        btnRegistrar.setText("Registrar un proyecto");
        btnRegistrar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnRegistrarActionPerformed(evt);
            }
        });
        getContentPane().add(btnRegistrar, new
org.netbeans.lib.awtextra.AbsoluteConstraints(580, 290, 150, -1));

        btnBorrar.setBackground(new java.awt.Color(0, 51, 51));
        btnBorrar.setForeground(new java.awt.Color(204, 204, 204));
        btnBorrar.setText("Borrar proyecto");
        btnBorrar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnBorrarActionPerformed(evt);
            }
        });
        getContentPane().add(btnBorrar, new
org.netbeans.lib.awtextra.AbsoluteConstraints(580, 340, 150, -1));

```

```

    }
});
getContentPane().add(btnBorrar, new
org.netbeans.lib.awtextra.AbsoluteConstraints(580, 340, 150, -1));

txtNombre.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
getContentPane().add(txtNombre, new
org.netbeans.lib.awtextra.AbsoluteConstraints(580, 110, 120, -1));

txtResponsable.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtResponsableActionPerformed(evt);
    }
});
getContentPane().add(txtResponsable, new
org.netbeans.lib.awtextra.AbsoluteConstraints(580, 170, 120, -1));

txtEstado.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtEstadoActionPerformed(evt);
    }
});
getContentPane().add(txtEstado, new
org.netbeans.lib.awtextra.AbsoluteConstraints(580, 230, 120, -1));

jLabel1.setText("Nombre del Proyecto");
getContentPane().add(jLabel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(580, 90, 160, -1));

jLabel2.setText("Trabajador responsable");
getContentPane().add(jLabel2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(580, 150, 160, -1));

jLabel3.setText("Estado");

```

```

        getContentPane().add(jLabel3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(580, 210, 140, -1));

        TableProyecto.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
                "Nombre Proyecto", "Trabajador ", "Estado"
            }
        ));
        jScrollPane2.setViewportViewView(TableProyecto);

        getContentPane().add(jScrollPane2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 70, 570, 330));

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

    private void txtResponsableActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void txtEstadoActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void btnRegistrarActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            String nombre = txtNombre.getText();
            String responsable = txtResponsable.getText();
            String estado = txtEstado.getText();
            // Validar si alguno de los campos está vacío
            if (nombre.isEmpty() || responsable.isEmpty() || estado.isEmpty()) {

```

```

        // Lanzar una excepción si los campos están vacíos
        JOptionPane.showMessageDialog(null, "Por favor, complete todos los campos.");
    }
    model.addRow(new Object[]{nombre, responsable, estado});
    // Limpiar los campos de texto después de registrar
    txtNombre.setText("");
    txtResponsable.setText("");
    txtEstado.setText("");
} catch (HeadlessException e){
    JOptionPane.showMessageDialog(null, "Debe ingresar obligatoriamente texto en
    todos los TXT.");

}

}

private void btnBorrarActionPerformed(java.awt.event.ActionEvent evt) {
    int rowCount = model.getRowCount();
    if (rowCount > 0) { // Si hay datos en la tabla
        // Confirmación para evitar borrado accidental
        int confirmacion = JOptionPane.showConfirmDialog(
            null,
            "¿Estás seguro de que quieres borrar todos los datos?",
            "Confirmación",
            JOptionPane.YES_NO_OPTION
        );
        if (confirmacion == JOptionPane.YES_OPTION) {
            // Eliminar todas las filas
            model.setRowCount(0);
            JOptionPane.showMessageDialog(null, "Todos los datos han sido borrados.");
        }
    } else {
        JOptionPane.showMessageDialog(null, "No hay datos para borrar.");
    }
}
}

```



```

/**
 * @param args the command line arguments
 */
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(interfaz.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

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

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

```

```
java.util.logging.Logger.getLogger(interfaz.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
}
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

```
java.awt.EventQueue.invokeLater(() -> {
```

```
    new interfaz().setVisible(true);
```

```
});
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JTable TableProyecto;
```

```
private javax.swing.JButton btnBorrar;
```

```
public javax.swing.JButton btnRegistrar;
```

```
private javax.swing.JLabel jLabel1;
```

```
private javax.swing.JLabel jLabel2;
```

```
private javax.swing.JLabel jLabel3;
```

```
private javax.swing.JPanel jPanel1;
```

```
private javax.swing.JPanel jPanel2;
```

```
private javax.swing.JScrollPane jScrollPane2;
```

```
public javax.swing.JTextField txtEstado;
```

```
public javax.swing.JTextField txtNombre;
```

```
public javax.swing.JTextField txtResponsable;
```

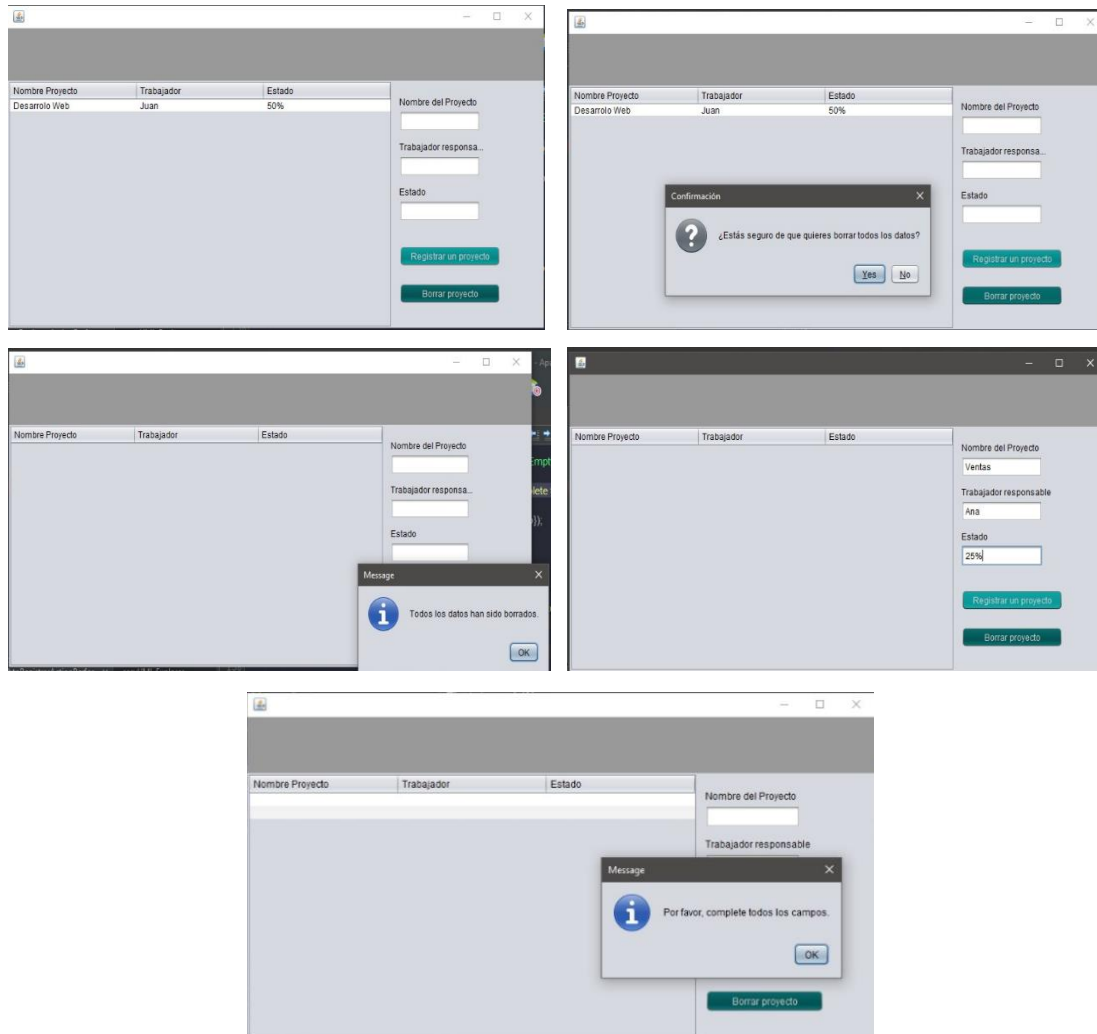
```
// End of variables declaration
```

```
}
```

Se adjunta el link del repositorio donde se aloja el proyecto:

<https://github.com/LoorJair/Actividad-Experimental>

3. Salidas Del Código



Conclusión:

La implementación del sistema de gestión de proyectos utilizando el paradigma orientado a objetos fue efectiva y pudimos cumplir con los objetivos establecidos, proyecto desarrollado facilita el registro de proyectos, la asignación de tareas y la visualización del estado de cada uno, asegurando una organización clara y estructurada, la utilización de POO brindó importantes ventajas, como modularidad y la reutilización del código, además, la separación de responsabilidades entre clases y métodos contribuyó a que el sistema sea fácil de mantener y entender, sentando una base sólida para futuras modificaciones, por otro lado, la implementación de funcionalidades como la asignación de tareas y el seguimiento del estado de los proyectos fue práctica y efectiva. El sistema cumplió con las expectativas al proporcionar una herramienta básica pero funcional para gestionar proyectos de manera eficiente.

Bibliografía

¿Para que sirve el try y catch? en java. (s. f.). Stack Overflow En Español.

<https://es.stackoverflow.com/questions/96278/para-que-sirve-el-try-y-catch-en-java>

Punto y Coma. (2020, 29 marzo). java archivos de texto y Jtable [Video]. YouTube.

<https://www.youtube.com/watch?v=re1V7YdoVcs>