

UNIDAD 12

MODELO VISTA CONTROLADOR

**Programación
CFGS DAW**

Paco Aldarias
paco.aldarias@ceedcv.es
Curso 2020/2021
30/03/21 18:49:23

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

▮ Importante

▮ Atención

▮ Interesante

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN.....	3
2. CONCEPTOS BÁSICOS.....	3
3. EJEMPLO MVC SIMPLE.....	4
3.1 Descripción.....	4
3.2 Controlador.....	5
3.3 Modelo.....	6
3.4 Vista.....	7
4. EJEMPLO MVC con swing.....	8
4.1 Descripción.....	8
4.2 Controlador.....	10
4.3 Modelo.....	14
4.4 Vista.....	14

UD12. MVC

1. INTRODUCCIÓN

Este documento esta realizado para poder explicar que es el modelo vista controlador (MVC) de forma simple.

El patrón MVC (Modelo, Vista, Controlador), es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz que ve el usuario de la logica del negocio en tres componentes distintos.

El MVC se puede utilizar en cualquier lenguaje de programación.

2. CONCEPTOS BÁSICOS

El MVC es una forma de realizar los programas, es decir, tiene que ver con el diseño estructural de programas.

Esta arquitectura establece que en la programación de aplicaciones con interfaz gráfica de usuario deben aparecer tres partes fundamentales.

Conceptualmente podemos considerar:

1.- Modelo (model). Parte que albergará los datos y cuya construcción será totalmente ajena a la forma en que se muestren o recogan los datos. Serian las clases Persona, Profesor pero sin operaciones de entrada salida como `println` o `scanner`, y sin operaciones de la lógica del programa.

2.- Vista (view). Serán las clases que permiten operaciones de entrada salida con el usuario. Suelen corresponder con las clases de Modelo para recogida datos o visualización de los mismos.

3.- Controlador (controller). Parte que se ocupará de la interpretación de las entradas del usuario, las cuales permiten modificar el modelo (datos) de forma adecuada. Sería las clases que controlan la lógica del programa.

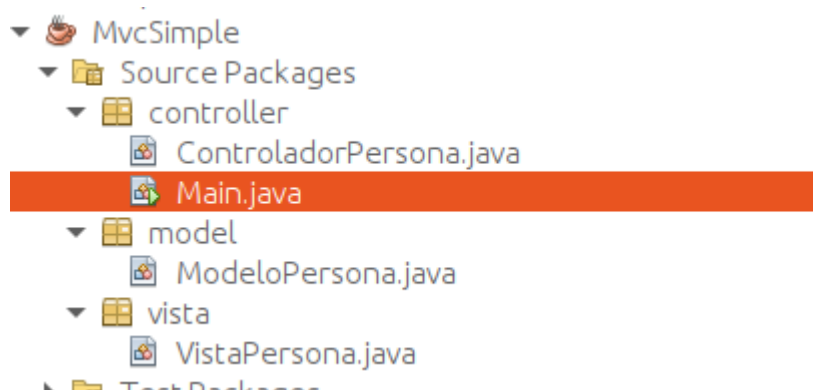
Para saber si está bien el MVC de una aplicación deberá cumplirse:

1. En el modelo no deben existir entradas salidas, es decir, no habrá `println`, si `scanner`. Esto ata el modelo de datos a donde se muestran los datos. Es decir, que podremos usar la terminal, `swing` o un fichero para comunicarnos con el usuario.
2. En el controlador no deben existir entradas salidas, es decir, no habrá `println`, si `scanner`. Por lo mismo de antes el controlador debe ser independiente de los datos y por donde se muestra o piden los datos.
3. En la vista no se describen datos, ni toma el control de la aplicación.

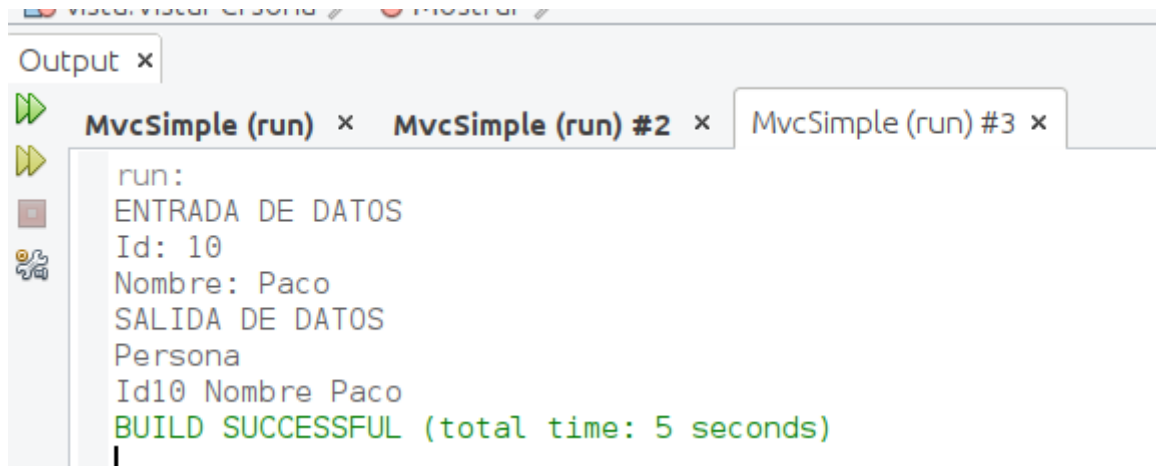
3. EJEMPLO MVC SIMPLE

3.1 Descripción

El siguiente ejemplo simple de un MVC, seria aquel que pide los datos y después los muestra en pantalla por terminal. Se organiza en tres paquetes diferentes.



Resultado de la ejecución:



3.2 Controlador

Esta dentro del paquete controller y no tiene operaciones de entrada salida, ni define datos. El main inicia el programa creando objetos para poder usar sus funciones. El Controller persona es el que tiene la lógica del programa utilizando objetos del Modelo y de la Vista.

Fichero: Main.java

```
package controller;
import model.ModeloPersona;
import vista.VistaPersona;

public static void main(String[] args) {
    ModeloPersona m = new ModeloPersona();
    VistaPersona v = new VistaPersona();
    ControladorPersona c = new ControladorPersona(m, v);
}

}
```

Fichero: ControladorPersona.java

```
package controller;
import model.ModeloPersona;
import vista.VistaPersona;

class ControladorPersona {

    ModeloPersona modelo;
```

```
VistaPersona vista;
```

```
ControladorPersona(ModeloPersona m, VistaPersona v) {  
    modelo = m;  
    vista = v;  
    inicia();  
}
```

```
private void inicia() {  
    modelo = vista.Pedir();  
    vista.Mostrar(modelo);  
  
}  
}
```

3.3 Modelo

Esta dentro del paquete modelo y no tiene operaciones de entrada salida, ni tiene lógica (bucles). En esta se define datos de nuestro diagrama de clases

Fichero: ModeloPersona.java

```
package model;
```

```
public class ModeloPersona {  
  
    private String id ;  
    private String nombre;  
  
    public String getId() {  
        return id;  
    }  
  
    public void setId(String id) {  
        this.id = id;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
}
```

```
public void setNombre(String nombre) {  
    this.nombre = nombre;  
}  
}
```

3.4 Vista

Se encuentra en el paquete vista y contiene aquellas clases que realizan entrada salida con el usuario. En este ejemplo la vista utiliza la terminal tanto para la entrada como la salida.

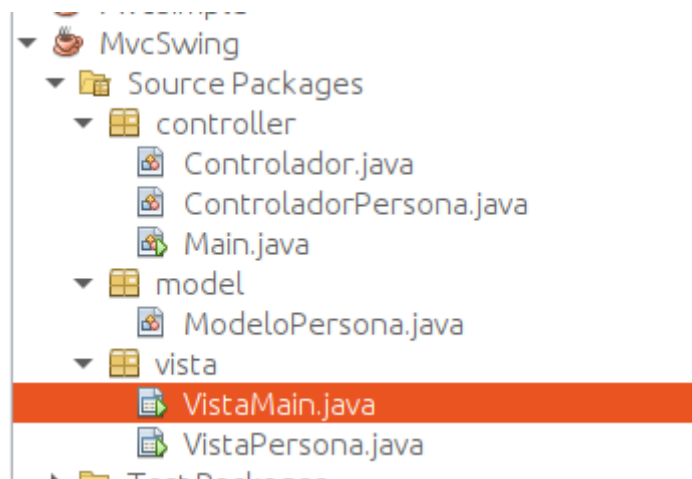
Fichero: VistaPersona.java

```
public void Mostrar( ModeloPersona p){  
  
    System.out.println("SALIDA DE DATOS" );  
    System.out.println("Persona");  
    System.out.println("Id" + p.getId() + " Nombre "+p.getNombre());  
}  
  
public ModeloPersona Pedir(){  
    Scanner s = new Scanner(System.in);  
    ModeloPersona p = new ModeloPersona();  
    String id;  
    String nombre;  
  
    System.out.println("ENTRADA DE DATOS" );  
    System.out.print("Id: " );  
    id = s.next();  
    p.setId(id);  
  
    System.out.print("Nombre: " );  
    nombre = s.next();  
    p.setNombre(nombre);  
  
    return p;  
}  
}
```

4. EJEMPLO MVC CON SWING

4.1 Descripción

En este ejemplo vamos a separar el MVC en carpetas como se ve en la imagen:



La aplicación tiene una pantalla inicial llamada VistaMain con 3 botones.



Tendremos la pantalla VistaPersona con los datos de una Persona. VistaPersona permitirá editar si entramos por el botón Pedir o sólo muestra los datos si pulsamos el botón Mostrar. Saldremos de la app pulsado Salir.

Desde le menú principal si se pulsa Pedir, tendremos que VistaPersona con los JTextField editables seria esta ventana:



Pedir

iD 001

Nombre Juan

Enviar

Al pulsar Enviar no salimos guardando los datos en un objeto del modelo Persona.

Desde le menú principal si se pulsa Mostrar, tendremos que VistaPersona con los JTextField no editables seria esta ventana:



Mostrar

iD 001

Nombre Juan

Enviar

4.2 Controlador

En el Main tendremos la función principal que llamara al controlador principal llamado Controlador. Controlador configura los botones y llama al ControladorPersona si se pulse los botones de Mostrar o Pedir. Y sale de la aplicación al pulsar Salir.

Fichero: Main.java

```
package controller;
import model.ModeloPersona;
import vista.VistaMain;
public class Main {
    public static void main(String[] args) {
        ModeloPersona m = new ModeloPersona();
        VistaMain v = new VistaMain();
        Controlador c = new Controlador(m, v);
    }
}
```

Fichero: Controlador.java

```
package controller;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import model.ModeloPersona;
import vista.VistaMain;

class Controlador implements ActionListener {

    ModeloPersona modelo;
    VistaMain vistaMain;
    ControladorPersona controladorPersona;

    Controlador(ModeloPersona m, VistaMain v) {
        modelo = m;
        vistaMain = v;
        inicia();
    }

    private void inicia() {
        vistaMain.setVisible(true);
        programaBotones();
        vistaMain.getJLabelTitulo().setText("Menu Principal");
    }
}
```

```
}

@Override
public void actionPerformed(ActionEvent ae) {

    String comando = ae.getActionCommand();

    switch (comando) {
        case "Mostrar":
            mostrar();
            break;
        case "Pedir":
            pedir();
            break;
        case "Salir":
            salir();
            break;
    }
}

private void programaBotones() {

    // Ponemos texto
    vistaMain.getJButtonMostrar().setActionCommand("Mostrar");
    vistaMain.getJButtonPedir().setActionCommand("Pedir");
    vistaMain.getJButtonSalir().setActionCommand("Salir");

    // Se pone a escuchar las acciones del usuario
    vistaMain.getJButtonMostrar().addActionListener(this);
    vistaMain.getJButtonPedir().addActionListener(this);
    vistaMain.getJButtonSalir().addActionListener(this);
}

private void mostrar() {

    controladorPersona = new ControladorPersona(modelo, vistaMain, "Mostrar");
}

private void pedir() {
    controladorPersona = new ControladorPersona(modelo, vistaMain, "Pedir");
}

private void salir() {
    System.exit(0);
}
```

```
}
```

Fichero: ControladorPersona.java

```
package controller;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import model.ModeloPersona;
import vista.VistaMain;
import vista.VistaPersona;

class ControladorPersona implements ActionListener {

    private ModeloPersona modelo;
    private VistaPersona vistaPersona;
    private VistaMain vistaMain;
    private String operacion;

    ControladorPersona(ModeloPersona m, VistaMain vm, String op) {
        modelo = m;
        vistaMain = vm;
        operacion = op;
        inicia();
    }

    private void inicia() {

        vistaMain.setVisible(false);
        vistaPersona = new VistaPersona();
        vistaPersona.setVisible(true);

        if (modelo.getId().length() > 0) {
            rellenaTexto();
        }
        programaBotones();
    }

    @Override
    public void actionPerformed(ActionEvent ae) {

        String comando = ae.getActionCommand();

        switch (comando) {
            case "Enviar":
```

```
        enviar();
        break;
    }
}

private void programaBotones() {

    // Ponemos el titulo
    vistaPersona.getjLabelTitulo().setText(operacion);

    // Ponemos texto
    vistaPersona.getjButtonEnviar().setActionCommand("Enviar");

    // Se pone a escuchar las acciones del usuario
    vistaPersona.getjButtonEnviar().addActionListener(this);

    if (operacion.compareTo("Mostrar") == 0) {
        vistaPersona.getjTextFieldId().setEditable(false);
        vistaPersona.getjTextFieldNombre().setEditable(false);
    }
}

private void enviar() {

    String id;
    String nombre;

    id = vistaPersona.getjTextFieldId().getText();
    nombre = vistaPersona.getjTextFieldNombre().getText();

    modelo.setId(id);
    modelo.setNombre(nombre);

    vistaPersona.setVisible(false);
    vistaPersona.dispose();
    vistaMain.setVisible(true);
}

private void rellenaTexto() {

    String id = new String(modelo.getId());
    String nombre = new String(modelo.getNombre());

    vistaPersona.getjTextFieldId().setText(id);
    vistaPersona.getjTextFieldNombre().setText(nombre);
}
```

```
}  
  
}
```

4.3 Modelo

Fichero: **ModeloPersona.java**

```
package model;  
  
public class ModeloPersona {  
  
    private String id;  
    private String nombre;  
  
    public ModeloPersona() {  
        this.id = "";  
        this.nombre = "";  
    }  
  
    public String getId() {  
        return id;  
    }  
  
    public void setId(String id) {  
        this.id = id;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
}
```

4.4 Vista

Las vistas se crean con el asistente y se crean al final del fichero funciones getX siendo X el nombre del elementos a obtener.

Fichero: VistaMain.java

```
package vista;

import javax.swing.JButton;

/**
 *
 * @author paco
 */
public class VistaMain extends javax.swing.JFrame {

    /**
     * Creates new form vistaMain
     */
    public VistaMain() {
        initComponents();
        setLocationRelativeTo(null); // Centramos la ventana
    }

    /**
     * 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();
        jButtonPedir = new javax.swing.JButton();
        jButtonMostrar = new javax.swing.JButton();
        jLabelTitulo = new javax.swing.JLabel();
        jButtonSalir = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jButtonPedir.setText("Pedir");

        jButtonMostrar.setText("Mostrar");

        jLabelTitulo.setText("MENU PRINCIPAL");

        jButtonSalir.setText("Salir");
    }
}
```

```

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
            .addGap(114, Short.MAX_VALUE)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabelTitulo)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                        .addComponent(jButtonPedir)
                        .addGap(76, 76, 76)
                        .addComponent(jButtonMostrar)
                        .addGap(99, 99, 99))
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                        .addComponent(jButtonSalir)
                        .addGap(178, 178, 178))))))
);
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(54, 54, 54)
            .addComponent(jLabelTitulo)
            .addGap(37, 37, 37)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jButtonPedir)
                .addComponent(jButtonMostrar))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 40,
Short.MAX_VALUE)
            .addComponent(jButtonSalir)
            .addGap(95, 95, 95))
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(

```



```
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

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

/**
 * @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(VistaMain.class.getName()).log(java.util.logging.Level.SEVERE
, null, ex);
    } catch (InstantiationException ex) {

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

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

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

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

```
        new VistaMain().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton jButtonMostrar;
private javax.swing.JButton jButtonPedir;
private javax.swing.JButton jButtonSalir;
private javax.swing.JLabel jLabelTitulo;
private javax.swing.JPanel jPanel1;
// End of variables declaration

public JButton getjButtonMostrar() {
    return jButtonMostrar;
}

public JButton getjButtonPedir() {
    return jButtonPedir;
}

public javax.swing.JLabel getjLabelTitulo() {
    return jLabelTitulo;
}

public javax.swing.JButton getjButtonSalir() {
    return jButtonSalir;
}
}
```

Fichero: VistaPersona.java

```
package vista;

public class VistaPersona extends javax.swing.JFrame {

    public VistaPersona() {
        initComponents();
        setLocationRelativeTo(null);
    }

    /**
     * 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();
    jTextField1 = new javax.swing.JTextField();
    jLabelTitulo = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jTextFieldNombre = new javax.swing.JTextField();
    jButtonEnviar = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jTextField1.setText("001");

    jLabelTitulo.setText("VISTA PERSONA");

    jLabel2.setText("iD");

    jLabel3.setText("Nombre");

    jTextFieldNombre.setText("Juan");

    jButtonEnviar.setText("Enviar");

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jTextField1)
                    .addComponent(jLabelTitulo)
                    .addComponent(jLabel2)
                    .addComponent(jLabel3)
                    .addComponent(jTextFieldNombre)
                    .addComponent(jButtonEnviar, javax.swing.GroupLayout.DEFAULT_SIZE, 100, true))
                .addContainerGap(100, true))
    );
    jPanel1Layout.setVerticalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabelTitulo)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel2)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel3)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jTextFieldNombre, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButtonEnviar)
                .addContainerGap(100, true))
    );
}

```

```

        .addComponent(jTextFieldNombre, javax.swing.GroupLayout.PREFERRED_SIZE,
145, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(jPanel1Layout.createSequentialGroup())
        .addGap(138, 138, 138)
        .addComponent(jLabelTitulo)))
        .addContainerGap(41, Short.MAX_VALUE))
    );
    jPanel1Layout.setVerticalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup())
        .addGap(28, 28, 28)
        .addComponent(jLabelTitulo)
        .addGap(18, 18, 18)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(jTextFieldId, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel2))
        .addGap(40, 40, 40)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel3)
            .addComponent(jTextFieldNombre, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 42,
Short.MAX_VALUE)
        .addComponent(jButtonEnviar)
        .addGap(56, 56, 56))
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    pack();
}

```

```
/**
 * @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(VistaPersona.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

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

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

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

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

// Variables declaration - do not modify
private javax.swing.JButton jButtonEnviar;
```

```
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabelTitulo;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JTextField jTextFieldNombre;  
private javax.swing.JTextField jTextFieldId;  
// End of variables declaration  
  
public javax.swing.JButton getJButtonEnviar() {  
    return jButtonEnviar;  
}  
  
public javax.swing.JTextField getJTextFieldNombre() {  
    return jTextFieldNombre;  
}  
  
public javax.swing.JTextField getJTextFieldId() {  
    return jTextFieldId;  
}  
  
public javax.swing.JLabel getJLabelTitulo() {  
    return jLabelTitulo;  
}  
}
```

Licencia



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.