



**INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO**

“Practica 5 – Uso de GUI”

-Tinoco Videgaray Sergio Ernesto

Grupo: 3BV1

Materia: Paradigmas de programación

INSTITUTO POLITÉCNICO NACIONAL



ESCOM®

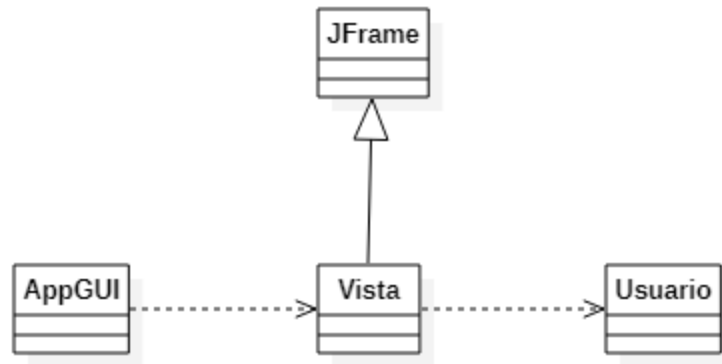
06/11/21

En el desarrollo de esta práctica se va a desarrollar un programa con una interfaz gráfica en la cual se van a capturar los datos del usuario como el nombre, correo, teléfono y contraseña del usuario para posteriormente procesarlos y mostrarlos en una ventana emergente.

Para esto se va a desarrollar una clase Vista que hereda de la clase JFrame para el manejo de los elementos gráficos como los botones y las cajas de texto.

Además se va a desarrollar una clase Usuario para almacenar los datos del usuario capturados por medio de las cajas de texto de la interfaz gráfica.

Considerando el siguiente diagrama de clases:



Primeramente se va a definir la clase Vista:

```
public class Vista extends JFrame {

    // DECLARACION DE CAMPOS Y BOTONES

    private final JTextField jTextEmail = new JTextField(),
        jtxtName = new JTextField(),
        jtxtMobilePhone = new JTextField(),
        jtxtPhoto = new JTextField();
    private final JPasswordField jTextPassword = new JPasswordField(),
        jtxtConfirmPassword = new JPasswordField();

    // Cuadro Imagen
    private final JLabel jLabelFotografia = new JLabel();

    // Botones
    private final JButton jButtonProcesar = new JButton("Process");
    private final JButton jButtonLimpiar = new JButton("Clear");

    public Vista() // Constructor
    {
        this.setTitle("User Registration");
        this.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        this.setBounds(50, 50, 640, 480);

        initComponents();

        this.setVisible(true);
    }
}
```

Se declaran las variables de los componentes que se van a utilizar en nuestro frame.

Se le asignan los nombres y los valores de nuestros botones "Process" y "Clear"

Se declara un constructor perteneciente a la clase Vista para inicializar los valores del título de la ventana del frame y el tamaño inicial del frame.

Se invoca al método initComponents para generar cada componente dentro del frame actual.

Se habilita nuestro frame para que aparezca en la ventana de nuestro sistema.

```

public void initComponents()
{
    // Declaración de etiquetas
    JLabel etiquetaEmail    = new JLabel("E-Mail Address:");
    JLabel etiquetaPassword = new JLabel("Password:");
    JLabel etqName          = new JLabel("Name:");
    JLabel etqMobileNumber  = new JLabel("Mobile Number:");
    JLabel etqPhoto         = new JLabel("Photo:");
    JLabel etqConfirmPassword = new JLabel("Condirm Password:");

    // Diseña menú
    JMenuBar barraMenus = new JMenuBar();
    JMenu archivo      = new JMenu("File");
    JMenuItem salir    = new JMenuItem("Exit");
    this.setJMenuBar(barraMenus);
    barraMenus.add(archivo);
    archivo.add(salir);
}

```

Se declara un método para declarar e inicializar cada componente que estará en nuestro frame.

Se declaran las etiquetas de cada caja de texto.

Se define una barra de menú que contendrá los componentes de un submenú para salir

```

//Establecer coordenadas
etqName.setBounds(80,30,120,30);
jtxtName.setBounds(130,30,200,30);

etiquetaEmail.setBounds(28,80,120,30);
jTextEmail.setBounds(130,80,200,30);

etqMobileNumber.setBounds(28,130,120,30);
jtxtMobilePhone.setBounds(130,130,200,30);

etqPhoto.setBounds(80,180,70,30);
jtxtPhoto.setBounds(130,180,200,30);

etiquetaPassword.setBounds(60,230,70,30);
jTextPassword.setBounds(130,230,200,30);

etqConfirmPassword.setBounds(10,280,150,30);
jtxtConfirmPassword.setBounds(130,280,200,30);
//

```

Se establecen las posiciones y el tamaño de los componentes dentro del frame Vista.

```
jLabelFotografia.setBounds(380,70,180,200);
jButtonProcesar.setBounds(200,350,100,40);
jButtonLimpiar.setBounds(320,350,100,40);

// Propiedades de los componentes

jLabelFotografia.setBorder(BorderFactory.createLineBorder(Color.black));
```

Se definen las posiciones y el tamaño de los botones y el label dentro del frame.

```
this.add(jLabelFotografia);

this.add(etiquetaEmail);
this.add(jTextEmail);

this.add(etiquetaPassword);
this.add(jTextPassword);

this.add(etqName);
this.add(jtxtName);

this.add(etqPhoto);
this.add(jtxtPhoto);

this.add(etqConfirmPassword);
this.add(jtxtConfirmPassword);

this.add(etqMobileNumber);
this.add(jtxtMobilePhone);

this.add(jButtonProcesar);
this.add(jButtonLimpiar);
```

Agregación de los componentes al frame actual

```
// GESTION DE EVENTOS

// Lo que se debe hacer cuando el usuario elija la opción "Salir"
salir.addActionListener(evt -> gestionaSalir(evt));

// Lo que se debe hacer cuando el usuario elija el botón "Procesar"
jButtonProcesar.addActionListener(evt -> gestionaProcesar(evt));

// Lo que se debe hacer cuando el usuario elija la opción "Limpiar"
jButtonLimpiar.addActionListener(evt -> gestionaLimpiar(evt));
```

Asignación de listener's a cada botón de la interfaz para que en caso de que el botón sea pulsado se mande a llamar al método correspondiente.

Metodos para salir del programa que va a desplegar una ventana emergente preguntandole al usuario si desea salir del programa al momento de oprimir el boton “X” en la esquina superior derecha de la interfaz grafica, o bien pulsando la opcion de “Exit” que se despliega en el submenu de la parte superior.

En caso de que el usuario seleccione la opcion “YES” el programa manda a llamar al metodo System.exit que va a terminar la ejecucion de la maquina virtual de java cerrando el programa.

```
class MyWindowAdapter extends WindowAdapter
{
    @Override
    public void windowClosing(WindowEvent e)
    {
        exit();
    }
}
addWindowListener(new MyWindowAdapter());
```

```
public void gestionaSalir(java.awt.event.ActionEvent evt)
{
    exit();
}

public void exit()
{
    int respuesta = JOptionPane.showConfirmDialog(rootPane, "Desea salir?", "Aviso", JOptionPane.YES_NO_OPTION);
    if(respuesta==JOptionPane.YES_OPTION) System.exit(0);
}
```

Se declara el método `gestionaProcesar` para que una vez capturados los datos del usuario en las cajas de texto, se van a almacenar en la clase `usuario`. En caso de que alguno de los campos este vacío a la hora de procesarlos el programa mostrara una ventana emergente pidiéndole al usuario llenar todos los campos. De igual forma en caso de que las contraseñas ingresadas en los dos últimos campos no coincidan el programa mostrara un mensaje de Error indicando que las contraseñas no son las mismas. En caso contrario se van a procesar los datos capturados y se van a postrar en una ventana de dialogo.

```
public void gestionaProcesar(java.awt.event.ActionEvent evt)
{
    Usuario user=new Usuario();

    if(jtxtName.getText().equals("") || jtxtEmail.getText().equals("")
    || jtxtMobilePhone.getText().equals("") || jtxtPhoto.getText().equals("")
    || jtxtPassword.getText().equals("") || jtxtConfirmPassword.getText().equals("")){
        JOptionPane.showMessageDialog(rootPane,"Por favor rellena todos los campos","Faltan datos", JOptionPane.ERROR_MESSAGE);
    }
    else{
        if(!jtxtPassword.getText().equals(jtxtConfirmPassword.getText())){
            JOptionPane.showMessageDialog(rootPane,"Las contraseñas no coinciden","Error", JOptionPane.ERROR_MESSAGE);
        }
        else{
            jLabelFotografia.setIcon(new ImageIcon("img/"+jtxtPhoto.getText()+".png"));
            user.setNombre(jtxtName.getText());
            user.setCorreo(jtxtEmail.getText());
            user.setNumero(jtxtMobilePhone.getText());
            user.setContraseña(jtxtPassword.getText());

            String datosCaptados;

            datosCaptados = "DATOS DEL USUARIO\n";
            datosCaptados = datosCaptados.concat(user.getDatos());

            JOptionPane.showMessageDialog(rootPane, datosCaptados);
        }
    }
}
```

```

public void gestionaLimpiar(java.awt.event.ActionEvent evt)
{
    jtxtName.setText("");
    jTextEmail.setText("");
    jtxtMobilePhone.setText("");
    jtxtPhoto.setText("");
    jTextPassword.setText("");
    jtxtConfirmPassword.setText("");
    jLabelFotografia.setIcon(null);
}

```

Método para vaciar las cajas de texto y el cuadro de la imagen.

Posteriormente se declara la clase Usuario

```

package appgui;

/**
 *
 * @author Sergio
 */
public class Usuario {

    private String nombre, correo, numero, contraseña;

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

    public void setNumero(String num) {
        this.numero=num;
    }

    public void setCorreo(String email) {
        this.correo=email;
    }

    public void setContraseña(String pasw) {
        this.contraseña=pasw;
    }
}

```

Se declaran los métodos de la clase usuario para asignar los valores de los datos del usuario a las variables de la clase.

```

public String getDatos() {
    String datos;

    datos="\nNombre: "+this.nombre+
        "\nE-mail: "+this.correo+
        "\nNumero celular: "+this.numero;

    return datos;
}

```

Se define un método getDatos para recuperar los datos almacenados en las variables de la clase usuario previamente capturados y procesados.

Finalmente se declara la clase AppGUI la cual será la clase main

```
package appgui;

/**
 *
 * @author Sergio Tinoco
 * 06/11/21
 * /
 */
public class AppGui {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        new Vista();
    }

}
```

Se instancia un objeto anónimo de la clase Vista para generar los elementos de la interfaz de usuario.

Resultados de la ejecución:

The screenshot shows the 'User Registration' window with the following fields: Name (Peter Parker), E-Mail Address (spbenpark221@outlook.com), Mobile Number (718-518), Photo (peter_pa), Password (masked), and Confirm Password (masked). An 'Error' dialog box is displayed in the center with the message 'Las contraseñas no coinciden' (Passwords do not match). The 'Process' and 'Clear' buttons are at the bottom.

The screenshot shows the 'User Registration' window with the same fields as the previous one. A 'Message' dialog box is displayed in the center with the title 'DATOS DEL USUARIO' and the following information: Nombre: Peter Parker, E-mail: spbenpark221@outlook.com, and Numero celular: 718-5189. The 'Process' and 'Clear' buttons are at the bottom.

The screenshot shows the 'User Registration' window with the following fields: Name (John117), E-Mail Address (j117sp@hotmail.com), Mobile Number (854-117), Photo (master_), Password (masked), and Confirm Password (masked). A 'Faltan datos' (Missing data) dialog box is displayed in the center with the message 'Por favor rellena todos los campos' (Please fill in all fields). The 'Process' and 'Clear' buttons are at the bottom.

The screenshot shows the 'User Registration' window with the same fields as the previous one. A 'Message' dialog box is displayed in the center with the title 'DATOS DEL USUARIO' and the following information: Nombre: John117, E-mail: j117sp@hotmail.com, and Numero celular: 854-117. The 'Process' and 'Clear' buttons are at the bottom.

Observaciones y conclusiones:

El uso de interfaces graficas de usuario o “GUI” hacen más dinámico el proceso de captura de datos para posteriormente procesarlos, además de que son más atractivas para el usuario en comparación con las consolas que serían sus antecesoras. De igual forma el uso de clases para las GUI hace más factible el manejo de datos lo cual es un aspecto clave a la hora de implementar estos programas en un sistema con base de datos.