



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

MÓDULO 5: INTERFAZ GRÁFICA

Creación de los elementos gráficos y de interacción de los páneos

Esta guía es la continuación del reto 2 de este módulo. Aquí encontrará los pasos que siguen a lo que usted ya realizó.

ETAPA 1: PREPARACIÓN DEL ENTORNO DE DESARROLLO

Para preparar el entorno de desarrollo del reto siga la siguiente instrucción:

1. Abra en eclipse el proyecto en el que usted avanzó en el reto anterior (Reto: Cree usted mismo los páneos de la ventana principal de la aplicación)

ETAPA 2: IMPLEMENTACIÓN DE LA CLASE PANELDATOSPACIENTE

1. Recuerde que en el reto anterior, se asoció a la clase `PanelDatosPaciente` un distribuidor gráfico (layout) de tipo `BorderLayout`, mediante la siguiente instrucción en el método constructor de la clase:

```
setLayout( new BorderLayout( ) );
```

2. Ahora bien, se desea que la información del paciente se muestre de la siguiente manera: a la izquierda del panel, se deben mostrar los datos del paciente y a la derecha del mismo, su foto. Esto es imposible de lograr con un solo `BorderLayout` y/o con un `GridLayout`. Para resolver este problema, utilice dos páneos auxiliares, los cuales usará únicamente para agrupar la información de la forma deseada. Se usará un panel para agrupar los datos del paciente (`panelInformacion`) y otro para la foto (`panelFoto`), como se ilustra en la siguiente figura:



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java



Figura 14. Uso de los paneles auxiliares panelInformación y panelFoto dentro de PanelDatosPaciente para agrupar la información adecuadamente.

3. No es necesario que estos paneles auxiliares (panelInformacion y panelFoto) sean nuevas clases del programa. Basta con que los declaremos como variables locales de tipo JPanel en el método constructor de la clase PanelDatosPaciente. Añada panelInformacion en la posición CENTER y panelFoto en la posición EAST del panel. A continuación se muestran las instrucciones del método constructor de PanelDatosPaciente hasta este punto:

```
public class PanelDatosPaciente extends JPanel
{
    public PanelDatosPaciente()
    {
        setLayout( new BorderLayout( ) );
        TitledBorder border = BorderFactory.createTitledBorder("Datos del paciente");
        border.setTitleColor( Color.BLUE );
        setBorder( border );

        //Panel donde se muestra la foto del paciente
        JPanel panelFoto;

        // Panel donde se muestra la información del paciente
        JPanel panelInformacion;

        panelFoto = new JPanel( );
        add( panelFoto, BorderLayout.EAST );

        panelInformacion = new JPanel();
        add( panelInformacion, BorderLayout.CENTER );
    }
}
```

Figura 15. Método constructor de PanelDatosPaciente hasta este punto



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

4. Declare todos los elementos gráficos del panel como atributos de la clase `PanelDatosPaciente`:

```
public class PanelDatosPaciente extends JPanel
{
    //Etiqueta y zona de texto para el nombre
    private JLabel labNombre;
    private JTextField txtNombre;

    // Etiqueta y zona de texto para el apellido
    private JLabel labApellido;
    private JTextField txtApellido;

    // Etiqueta y zona de texto para la fecha de nacimiento
    private JLabel labFNacimiento;
    private JTextField txtFNacimiento;

    // Etiqueta y zona de texto para el sexo
    private JLabel labSexo;
    private JTextField txtSexo;

    // Etiqueta donde se muestra la imagen
    private JLabel labImagen;

    // Botón para calcular la edad
    private JButton butEdad;

    // Zona de texto donde se muestra la edad
    private JTextField txtEdad;
```

Figura 16. Declaración de los elementos gráficos del panel como atributos de la clase `PanelDatosPaciente`



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

5. Cree y configure todos los elementos gráficos del panel en el método constructor de la clase `PanelDatosPaciente`:

```
//Creación de todas las etiquetas
labNombre = new JLabel( "Nombre: " );
labApellido = new JLabel( "Apellido: " );
labSexo = new JLabel( "Sexo: " );
labFNacimiento = new JLabel( "Fecha de Nacimiento: " );

//Creación y configuración de la zona de texto para el nombre
txtNombre = new JTextField( 15 );
txtNombre.setEditable( false );
txtNombre.setBackground(Color.LIGHT_GRAY);
txtNombre.setForeground(Color.BLUE);

//Creación y configuración de la zona de texto para el apellido
txtApellido = new JTextField( 15 );
txtApellido.setEditable( false );
txtApellido.setBackground(Color.LIGHT_GRAY);
txtApellido.setForeground(Color.BLUE);

//Creación y configuración de la zona de texto para el sexo
txtSexo = new JTextField( 2 );
txtSexo.setEditable( false );
txtSexo.setBackground(Color.LIGHT_GRAY);
txtSexo.setForeground(Color.BLUE);

//Creación y configuración de la zona de texto para la fecha de nacimiento
txtFNacimiento = new JTextField( 10 );
txtFNacimiento.setEditable( false );
txtFNacimiento.setBackground(Color.LIGHT_GRAY);
txtFNacimiento.setForeground(Color.BLUE);

//Creación y configuración del botón y la zona de texto para la edad
butEdad = new JButton( "Calcular Edad" );
txtEdad = new JTextField( 10 );
txtEdad.setEditable( false );
```

Figura 17. Creación y configuración de los elementos gráficos del panel en el método constructor de la clase `PanelDatosPaciente`

6. Continuando con el método constructor de la clase `PanelDatosPaciente`, asocie al panel auxiliar `panelInformacion` (el que usted mismo creó en el paso 3) un



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

distribuidor gráfico de tipo GridLayout de 6 filas y 2 columnas y añade los elementos gráficos que van dentro del mismo:

```
panelInformacion.setLayout(new GridLayout(6,2));  
panelInformacion.add(labNombre);  
panelInformacion.add(txtNombre);  
panelInformacion.add(labApellido);  
panelInformacion.add(txtApellido);  
panelInformacion.add(labSexo);  
panelInformacion.add(txtSexo);  
panelInformacion.add(labFNacimiento);  
panelInformacion.add(txtFNacimiento);  
panelInformacion.add(butEdad);  
panelInformacion.add(txtEdad);
```

Figura 18. Adición de los elementos gráficos en el panel auxiliar panelInformacion

7. Por último, en el mismo método constructor de la clase PanelDatosPaciente, cree la etiqueta donde se mostrará la imagen del paciente y añádala al panel auxiliar panelFoto (el que usted mismo creó en el paso 3). Dado que aún no se ha cargado la información de ningún paciente, ponga el texto "FOTO" a la etiqueta, simplemente para que pueda visualizarla al ejecutar el programa:

```
labImagen = new JLabel("FOTO" );  
panelFoto.add( labImagen );
```

Figura 19. Adición de la etiqueta de la imagen en el panel auxiliar panelFoto



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

8. Ejecute la aplicación. En este punto de la implementación, se debe desplegar la ventana principal con los elementos gráficos del panel de datos del paciente (sin ningún valor) y con los demás paneles vacíos, como se muestra en la siguiente figura:

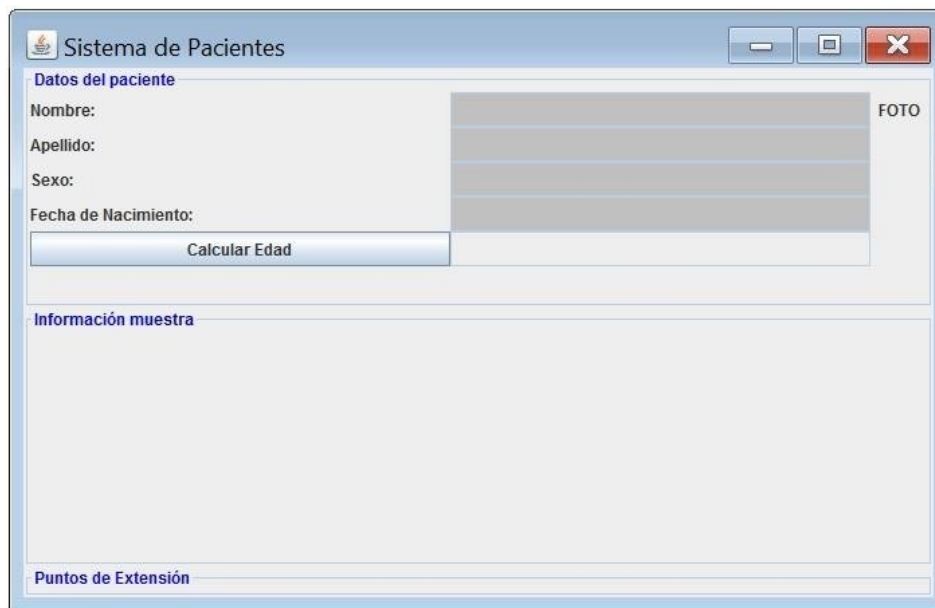


Figura 20. Apariencia de la ventana principal hasta este punto del desarrollo.

ETAPA 3: IMPLEMENTACIÓN DE LA CLASE PANELDATOSMUESTRA

Tomando como ejemplo los pasos que siguió en la etapa anterior para implementar la clase `PanelDatosPaciente`, es su turno de crear ahora la clase `PanelDatosMuestra`.

1. Recuerde que en el reto anterior, se asoció a la clase `PanelDatosMuestra` un distribuidor gráfico (layout) de tipo `BorderLayout`, mediante la siguiente instrucción en el método constructor de la clase:

```
setLayout( new BorderLayout( ) );
```



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

2. Este tipo distribuidor gráfico no es el más apropiado para la agrupación de los elementos gráficos de este panel. Cambie dicho distribuidor por uno de tipo `GridLayout`, de 6 filas y 4 columnas.
3. Declare todos los elementos gráficos del panel como atributos de la clase `PanelDatosMuestra`, siguiendo los tipos y nombres que se indican a continuación:

Elemento gráfico	Nombre del atributo en la clase <code>PanelDatosMuestra</code>	Descripción
<code>JLabel</code>	<code>labFTomaMuestra</code>	Etiqueta para la fecha de toma
<code>TextField</code>	<code>txtFTomaMuestra</code>	Zona de texto donde se ingresa la fecha de toma
<code>JLabel</code>	<code>labVolumenMuestra</code>	Etiqueta para el volumen
<code>TextField</code>	<code>txtVolumenMuestra</code>	Zona de texto donde se ingresa el volumen
<code>JLabel</code>	<code>labVolumenEritrocitos</code>	Etiqueta para el volumen de eritrocitos
<code>TextField</code>	<code>txtVolumenEritrocitos</code>	Zona de texto donde se ingresa el volumen de eritrocitos
<code>JLabel</code>	<code>labConteoLeucocitos</code>	Etiqueta para el conteo de leucocitos
<code>TextField</code>	<code>txtConteoLeucocitos</code>	Zona de texto donde se ingresa el conteo de leucocitos
<code>JLabel</code>	<code>labConteoPlaquetas</code>	Etiqueta para el conteo de plaquetas
<code>TextField</code>	<code>txtConteoPlaquetas</code>	Zona de texto donde se ingresa el conteo de plaquetas
<code>CheckBox</code>	<code>cbAyunas</code>	Caja de chequeo donde se indica si la muestra se tomó en ayunas
<code>Button</code>	<code>butHematocrito</code>	Botón para ejecutar el cálculo del valor de hematocrito
<code>TextField</code>	<code>txtHematocrito</code>	Zona de texto para mostrar el valor de hematocrito
<code>Button</code>	<code>butLeucocitos</code>	Botón para ejecutar el cálculo del valor de leucocitos
<code>TextField</code>	<code>txtLeucocitos</code>	Zona de texto para mostrar el valor de leucocitos

Figura 21. Tipos y nombres de los atributos de la clase `PanelDatosMuestra`.

4. Cree y configure todos los elementos gráficos del panel (los que acaba de declarar en el punto anterior) en el método constructor de la clase `PanelDatosMuestra`.
5. Añada al panel todos los elementos gráficos creados en el punto anterior. Aquí deberá emplear una astucia para poder organizar los elementos en la cuadrícula del distribuidor gráfico, que consiste en agregar etiquetas vacías en las casillas de las cuadrículas del `GridLayout` que se desean dejar vacías (es decir, sin un elemento gráfico particular), con la siguiente instrucción:

```
add (new JLabel (""));
```



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

La siguiente figura ilustra las casillas de la cuadrícula del GridLayout de la clase `PanelDatosMuestra` donde se deben añadir etiquetas vacías:

GridBagLayout
de 6 filas por 4
columnas

Sistema de Pacientes

Datos del paciente

Nombre: FOTO

Apellido:

Sexo:

Fecha de Nacimiento:

Información muestra

Fecha Toma Muestra:	<input type="text"/>	<input type="text"/>	<input type="checkbox"/> Ayunas
Volumen Muestra:	<input type="text"/>	<input type="text"/>	<input type="text"/>
Volumen Eritrocitos:	<input type="text"/>	<input type="button" value="Calcular Hematocrito"/>	<input type="text"/>
Conteo Leucocitos:	<input type="text"/>	<input type="button" value="Calcular Leucocitos"/>	<input type="text"/>
Conteo Plaquetas:	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Puntos de Extensión

Para dejar una «casilla vacía» dentro de la cuadrícula del GridBagLayout, se debe en realidad añadir una etiqueta con una cadena vacía.

Figura 22. Astucia para dejar casillas aparentemente vacías dentro de la cuadrícula de un GridLayout.

6. Ejecute la aplicación. En este punto de la implementación, se debe desplegar la ventana principal con los elementos gráficos del panel de datos de la muestra (sin ningún valor) y con el panel de extensión vacío, como se muestra en la siguiente figura:



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

Figura 23. Apariencia de la ventana principal hasta este punto del desarrollo.

ETAPA 4: IMPLEMENTACIÓN DE LA CLASE PANELEXTENSIONES

Tomando como ejemplo los pasos que siguió en la etapa anterior para implementar la clase `PanelDatosMuestra`, es su turno de crear ahora la clase `PanelExtensiones`.

1. Recuerde que en el reto anterior, se asoció a la clase `PanelExtensiones` un distribuidor gráfico (layout) de tipo `BorderLayout`, mediante la siguiente instrucción en el método constructor de la clase:

```
setLayout( new BorderLayout( ) );
```



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

- Este tipo distribuidor gráfico no es el más apropiado para la agrupación de los elementos gráficos de este panel. Se podría utilizar un distribuidor de tipo `GridLayout`, de 1 filas y 4 columnas. Sin embargo, le sugerimos asociar un layout de tipo `FlowLayout` para conservar la apariencia deseada. Para esto, puede usar la siguiente instrucción:

```
setLayout( new FlowLayout( ) );
```

- Declare todos los elementos gráficos del panel como atributos de la clase `PanelDatosMuestra`, siguiendo los tipos y nombres que se indican a continuación:

Elemento gráfico	Nombre del atributo en la clase <code>PanelDatosMuestra</code>	Descripción
<code>JButton</code>	<code>btnAvanzar</code>	Botón para avanzar al siguiente paciente
<code>JButton</code>	<code>btnRetroceder</code>	Botón para retroceder al paciente anterior
<code>JButton</code>	<code>butOpcion1</code>	Botón para la extensión 1
<code>JButton</code>	<code>butOpcion2</code>	Botón para la extensión 2

Figura 24. Tipos y nombres de los atributos de la clase `PanelExtensiones`.

- Ejecute la aplicación. En este punto de la implementación, se debe desplegar la ventana principal con los elementos gráficos de todos los paneles, como se muestra en la siguiente figura:



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

The screenshot shows a Java Swing window titled "Sistema de Pacientes". The window is divided into three main sections:

- Datos del paciente:** This section contains labels for "Nombre:", "Apellido:", "Sexo:", and "Fecha de Nacimiento:". To the right of these labels is a large, empty rectangular area labeled "FOTO". Below the "Fecha de Nacimiento:" label is a button labeled "Calcular Edad".
- Información muestra:** This section contains labels for "Fecha Toma Muestra:", "Volumen Muestra:", "Volumen Eritrocitos:", "Conteo Leucocitos:", and "Conteo Plaquetas:". To the right of the "Fecha Toma Muestra:" label is a checkbox labeled "Ayunas". Below the "Volumen Muestra:" label is a button labeled "Calcular Hematocrito". Below the "Conteo Leucocitos:" label is a button labeled "Calcular Leucocitos".
- Puntos de Extensión:** This section contains four buttons: "<", "Opción 1", "Opción 2", and ">".

Figura 25. Apariencia de la ventana principal hasta este punto del desarrollo.