



# INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

## MÓDULO 5: INTERFAZ GRÁFICA

### Creación de los paneles

Esta guía es la continuación del reto 1 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 la ventana principal de la aplicación)

#### ETAPA 2: IMPLEMENTACIÓN

1. Para cada uno de los paneles definidos en el bosquejo de la interfaz y el diagrama UML, cree un atributo en la clase de la ventana principal (Figura 8).

```
//-----  
// Atributos de la interfaz  
//-----  
  
/**  
 * Es el panel que contiene los datos del paciente  
 */  
private PanelDatosPaciente panelDatosPaciente;  
  
/**  
 * Es el panel que contiene los elementos para realizar consultas sobre paciente  
 */  
private PanelDatosMuestra panelDatosMuestra;  
  
/**  
 * Es el panel que contiene los elementos para ejecutar las extensiones y la navegación  
 */  
private PanelExtensiones panelExtensiones;
```

Figura 8. Paneles como atributos de la ventana principal.



## INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

2. En el método constructor de la clase **InterfazSistemaPacientes**, cree cada uno de los paneles y organícelos dentro de la interfaz, de acuerdo a su diseño.

```
/**
 * Construye una nueva interfaz inicializada con los
 * datos del sistema de pacientes. <br>
 */
public InterfazSistemaPacientes()
{
    setTitle( "Sistema de Pacientes" );
    setSize( 700, 450 );
    setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    setLayout( new BorderLayout( ) );

    // Crea la clase principal
    sistemaPacientes = new SistemaPacientes( );

    panelDatosPaciente = new PanelDatosPaciente();
    panelDatosMuestra = new PanelDatosMuestra();
    panelExtensiones = new PanelExtensiones();

    // organizar el panel principal
    add( panelDatosPaciente, BorderLayout.NORTH );
    add( panelDatosMuestra, BorderLayout.CENTER );
    add( panelExtensiones, BorderLayout.SOUTH );
}
```

Figura 9. Creación de los paneles en la ventana principal.

3. Siguiendo los mismos pasos descritos en la Figura 4, cree tres nuevas clases. Una para cada panel declarado como atributo. Esta vez NO seleccione la casilla “*public static void main (String[] args)*”

Las clases que corresponden a los paneles deben tener la extensión **JPanel**.



# INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

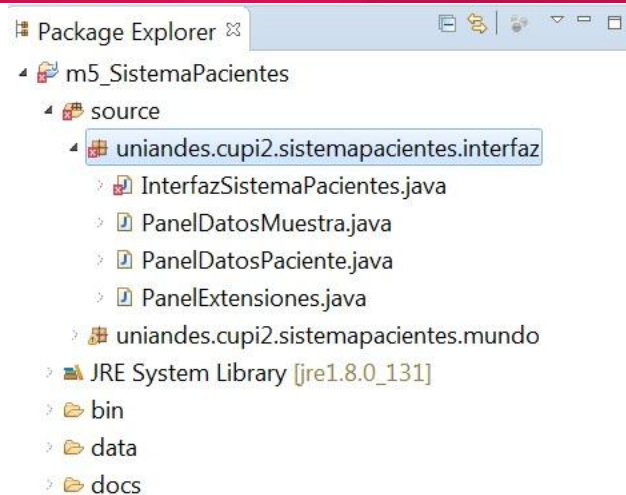


Figura 10. Creación de las clases correspondientes a los paneles.

4. En cada una de las nuevas clases, cree el constructor de acuerdo con su diseño inicial.

```
PanelDatosPaciente.java
1 package uniandes.cupi2.sistemapacientes.interfaz;
2
3 import java.awt.*;
4
5 import javax.swing.*;
6 import javax.swing.border.TitledBorder;
7
8 //-----
9 // Constructores
10 //-----
11
12 public class PanelDatosPaciente extends JPanel
13 {
14
15     public PanelDatosPaciente()
16     {
17         setLayout( new BorderLayout( ) );
18         TitledBorder border = BorderFactory.createTitledBorder("Datos del paciente");
19         border.setTitleColor( Color.BLUE );
20         setBorder( border );
21     }
22
23 }
```

Figura 11. Método constructor de la clase PanelDatosPaciente



# INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

```
PanelDatosMuestra.java
1 package uniandes.cupi2.sistemapacientes.interfaz;
2
3 import java.awt.*;
4
5 import javax.swing.*;
6 import javax.swing.border.TitledBorder;
7
8 //-----
9 // Constructores
10 //-----
11
12 public class PanelDatosMuestra extends JPanel
13 {
14
15     public PanelDatosMuestra()
16     {
17         setLayout( new BorderLayout( ) );
18         TitledBorder border = BorderFactory.createTitledBorder("Información muestra");
19         border.setTitleColor( Color.BLUE );
20         setBorder( border );
21     }
22
23 }
```

Figura 12. Método constructor de la clase PanelDatosMuestra

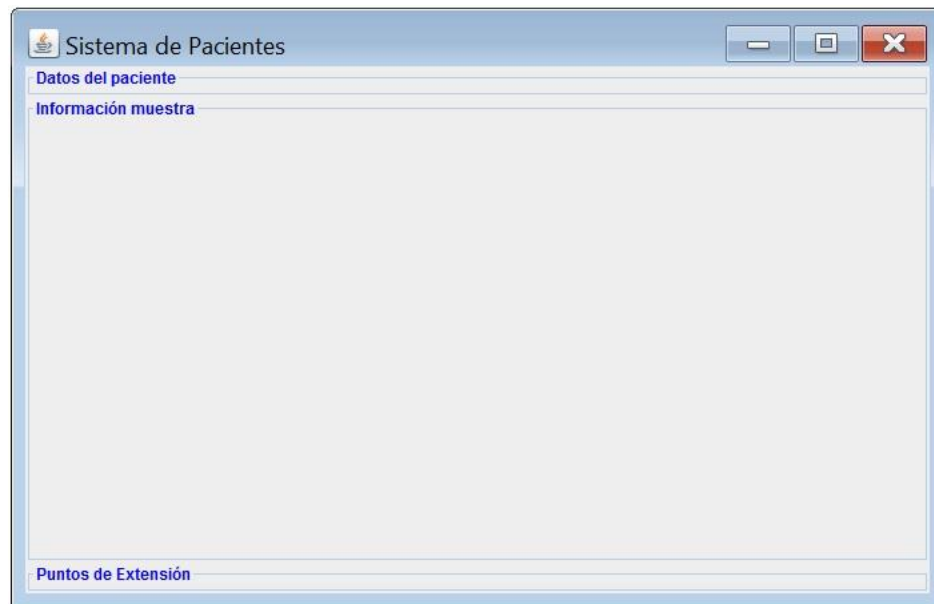
```
PanelExtensiones.java
1 package uniandes.cupi2.sistemapacientes.interfaz;
2
3 import java.awt.*;
4
5 import javax.swing.*;
6 import javax.swing.border.TitledBorder;
7
8 //-----
9 // Constructores
10 //-----
11
12 public class PanelExtensiones extends JPanel
13 {
14
15     public PanelExtensiones()
16     {
17         setLayout( new BorderLayout( ) );
18         TitledBorder border = BorderFactory.createTitledBorder("Puntos de Extensión");
19         border.setTitleColor( Color.BLUE );
20         setBorder( border );
21     }
22
23 }
```

Figura 13. Método constructor de la clase PanelExtensiones

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



# INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java



## RECOMENDACIÓN

Le recomendamos guardar todos los cambios efectuados al proyecto ya que en los siguientes retos de este módulo usted seguirá trabajando de manera incremental sobre este.