



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

MÓDULO 5: INTERFAZ GRÁFICA

Creación de la clase que representa la ventana principal y su relación con el mundo

ETAPA 1: PREPARACIÓN DEL ENTORNO DE DESARROLLO

Para preparar el entorno de desarrollo del reto siga las siguientes instrucciones:

1. Descargue el archivo comprimido **m5 SistemaPacientes** que contiene el proyecto.
2. Descomprima el archivo e impórtelo en el entorno Eclipse.

ETAPA 2: APROPIACIÓN DEL PROBLEMA

1. Lea la descripción del problema disponible en:
m5 SistemaPacientes/docs/specs/Descripcion.pdf
2. Lea el documento de requerimientos funcionales disponible en:
m5 SistemaPacientes/docs/specs/RequerimientosFuncionales.pdf
3. Estudie el modelo del mundo diseñado para este ejercicio. Este modelo se encuentra en: **m5 SistemaPacientes/docs/specs/ModeloConceptual**
4. Identifique las clases, relaciones entre clases, atributos y métodos.
5. Revise el proyecto Java. Identifique los elementos del mundo (clases, atributos y métodos).



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

ETAPA 3: IMPLEMENTACIÓN

Para crear la clase principal, primero es necesario crear el paquete que va a contener todas las clases de la interfaz.

1. Sobre la carpeta source del proyecto haga clic derecho y seleccione la opción de un nuevo paquete tal como se muestra en la Figura 2.

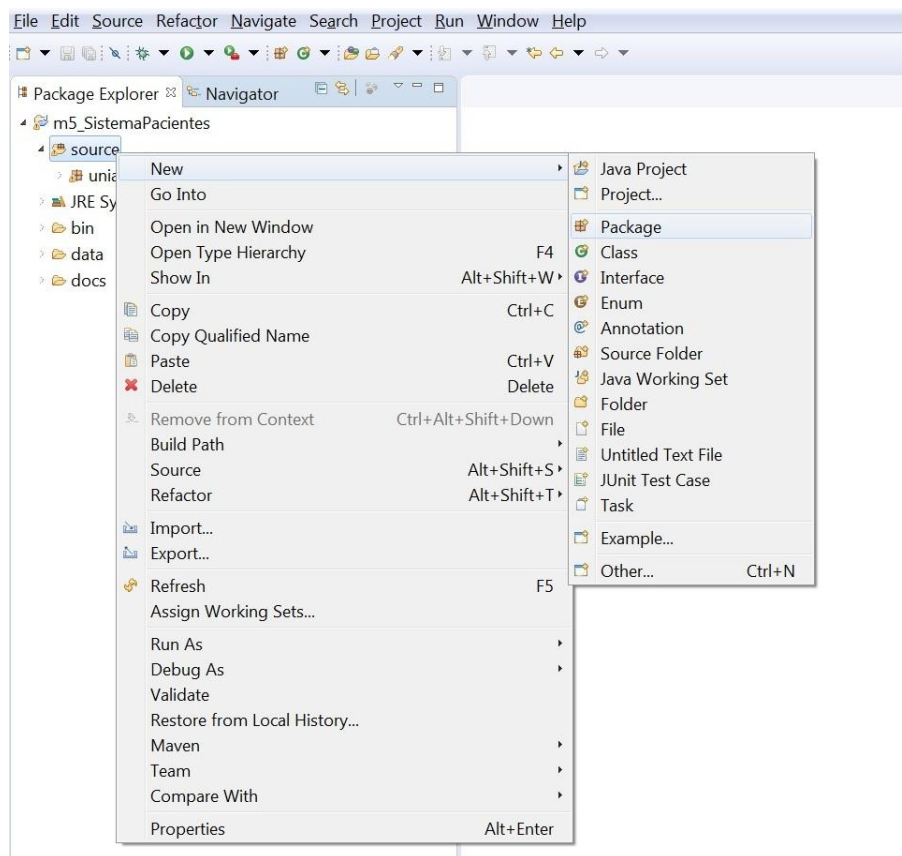


Figura 2. Creación de un nuevo paquete.



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

2. Cree el nuevo paquete con el nombre: uniandes.cupi2.sistemapacientes.interfaz

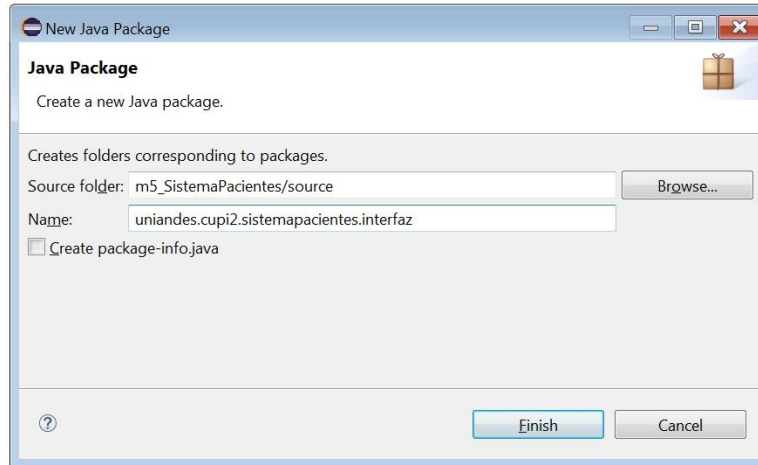


Figura 3. Creación de un nuevo paquete de interfaz.

Una vez creado el paquete de la interfaz, es hora de crear la clase principal de la interfaz.

1. Sobre el paquete que acaba de crear, haga click derecho y seleccione la opción de nueva clase, como se muestra en la Figura 4. El nombre de esta clase debe ser: **InterfazSistemaPacientes**.

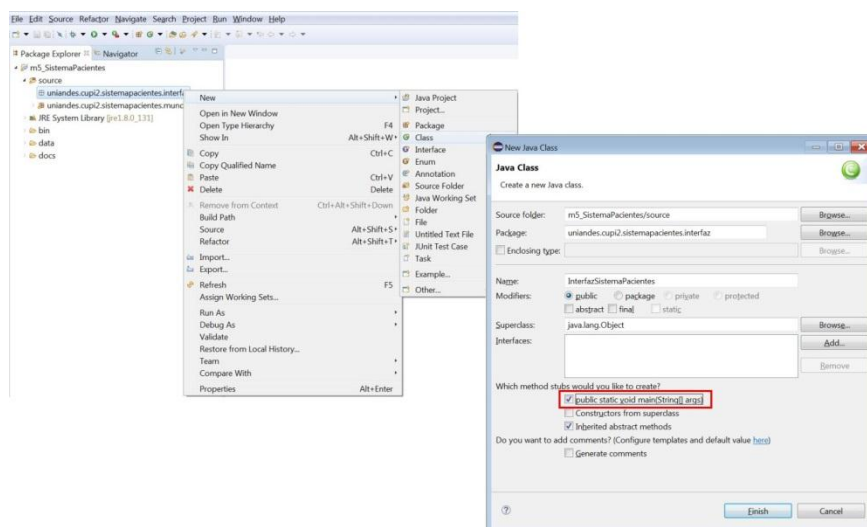


Figura 4. Creación de la clase principal de la interfaz.



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

2. Importe la biblioteca `javax.swing.*` y agregue la extensión `JFrame` a la clase nueva.

```
InterfazSistemaPacientes.java
1 package uniandes.cupi2.sistemapacientes.interfaz;
2
3 import javax.swing.*;
4
5 public class InterfazSistemaPacientes extends JFrame
```

Figura 5. Creación de la extensión `JFrame`.

3. Cree el método **main** en la clase principal de la interfaz.

```
InterfazSistemaPacientes.java
1 package uniandes.cupi2.sistemapacientes.interfaz;
2
3 import javax.swing.*;
4
5 public class InterfazSistemaPacientes extends JFrame
6 {
7
8     //-----
9     // Main
10    //-----
11
12    /**
13     * Este método ejecuta la aplicación, creando una nueva interfaz e inicializándola
14     * @param args Los argumentos no son utilizados.
15     */
16    public static void main(String[] args)
17    {
18        InterfazSistemaPacientes interfaz = new InterfazSistemaPacientes( );
19        interfaz.setVisible( true );
20    }
21 }
22
23 }
24
```

Figura 6. Método **main** de la clase `InterfazSistemaPacientes`.

4. Cree el método constructor con las características de la ventana principal, tal como se muestra en la Figura 7.



INTRODUCCIÓN A LA PROGRAMACIÓN por objetos en Java

5. Cree un atributo de la clase principal del mundo y cree el estado inicial del mundo en el constructor (Figura 7).

```
/**
 * Clase principal del mundo
 */
private SistemaPacientes sistemaPacientes;

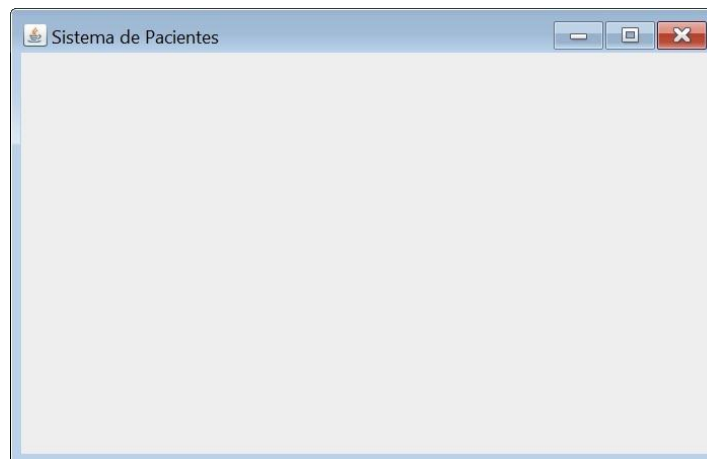
//-----
// Constructores
//-----

/**
 * 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( );
}
```

Figura 7. Constructor de la ventana principal.

6. Ejecute la aplicación. En este punto de la implementación, se debe desplegar la ventana principal vacía 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.