

Introducción al editor grafico de Jdeveloper

Las Interfaces graficas en java siempre me han parecido un poco fatídicas. Cuando he trabajado con awt, me ha parecido sencillo el construir el frame en general. Es decir, un frame con un panel, un par de botones con un textarea, etc. El problema esta cuando tienes que mover un botón y ponerlo mas a la izquierda y cosas por el estilo y te tienes que pelear con los Layout's. En Swing la cosa se complica un poco mas asi que creo que es necesario un buen editor grafico para java.

Sobretudo desde el punto de vista de una persona que al fin y al cabo es programador y no diseñador.

En este escrito tratare de explicar brevemente como conseguir una GUI en java de manera rápida con el editor grafico que acompaña al IDE Jdeveloper.

En primer lugar creamos una nueva área de trabajo, como "No template", donde se incluirá el nuevo proyecto. Para ello en Applications-Navigator click en botón-derecho "NewApplication workspace", se nos abrirá una ventana como la que se muestra a continuación en la figura 1.

Lo de no template lo haremos para que Jdeveloper no nos añada ningún patrón de diseño ni nada por el estilo. Nuestro objetivo es crear una clase java que tenga una serie de componentes gráficos y métodos vacíos que gestionen dichos componentes, que serán implementados por nosotros. Destacar que lo que queremos es crear una interfaz de forma rápida. Jdeveloper permite muchísimas mas cosas, pero no vamos a hablar de eso en este artículo.

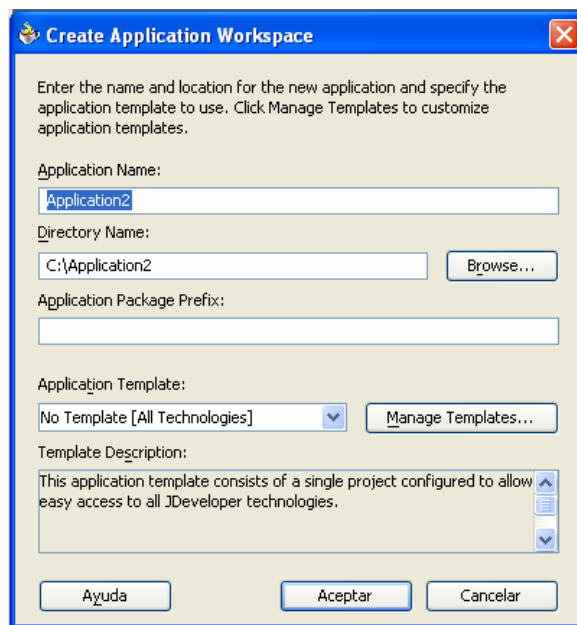


Figura 1

Una vez tenemos nuestro área de trabajo con nuestro proyecto .Sobre el proyecto con el botón-derecho y seleccionamos **New-->Client Tier-->Swing/awt-->Frame** Tal como se muestra en la figura 2.

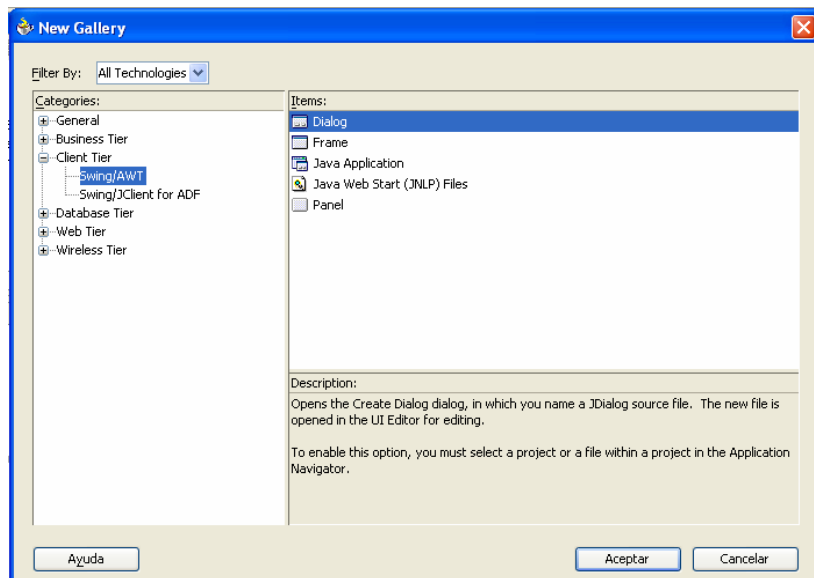


Figura 2

Como podemos ver esto ya nos ha creado una clase vacía la cual **extiende de JFrame**. También podemos observar que podemos ver la clase como diseño, como código y como clase.

Nos vamos al tipo de **vista diseño** y deberíamos ver algo como lo que se muestra en la figura 3.

Ahora solo se trata de añadir botones y componentes, como queramos nosotros, mediante el **Component palette** de Jdeveloper, si no tenemos activo el component lo activamos mediante **View--> component palette**.

En el component palette seleccionamos swing y simplemente se trata de seleccionar lo que queremos y ponerlo en nuestra interfaz.

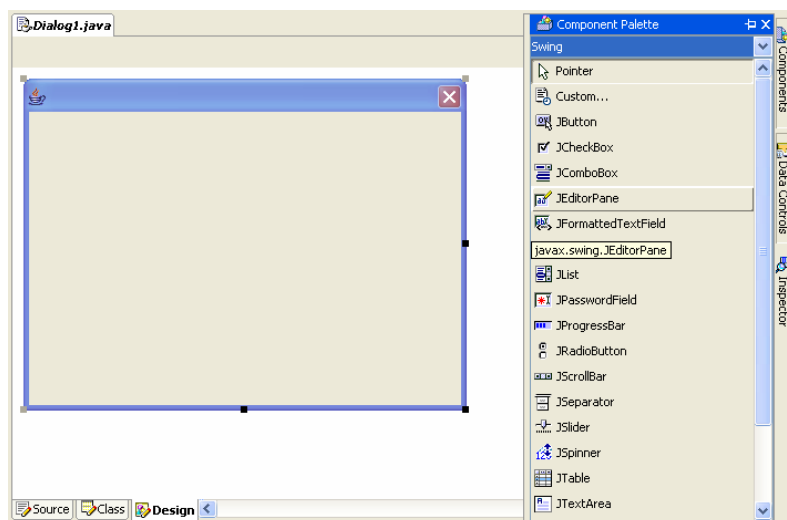


Figura 3

Yo para este ejemplo, he creado la interfaz que se ve a continuación. Como se puede ver es un textarea, un textfield y un botón. Muy parecido a lo que podría ser una pantalla de mensajería tipo Messenger.

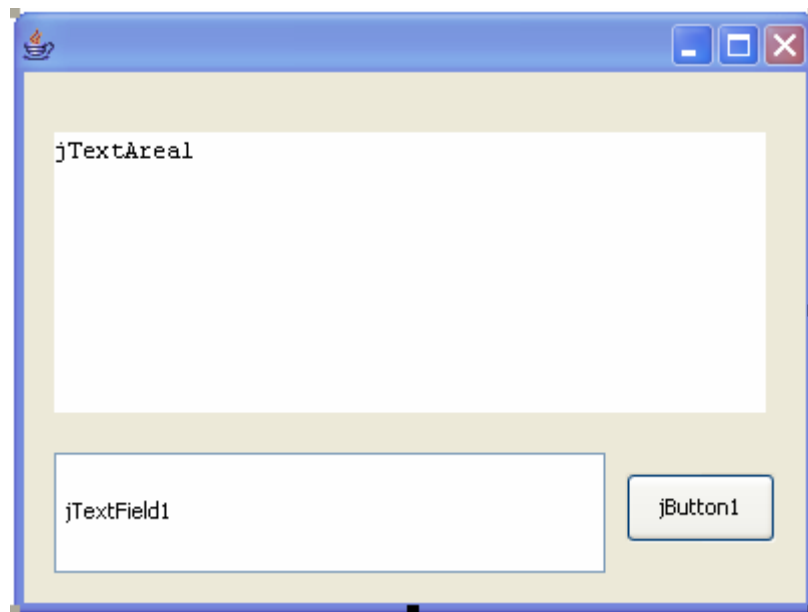


Figura 4

Si nos vamos al tipo **vista código** de nuestra clase podemos observar el código que ha generado Jdeveloper por nosotros. Hasta el momento esta interfaz no genera ningún evento y la clase java solo contiene una serie de componentes gráficos. El código es puro Swing así que no estamos atados a Jdeveloper para nada y podremos usar este código en cualquier otro IDE.

Ahora trataremos de gestionar los eventos que provocara la interacción del usuario con nuestra GUI. Para ello deberemos tener activado el **Property Inspector**, sino es así lo activamos mediante **View--> Property Inspector**.

Imaginemos que lo que queremos es añadir el método que gestionara los eventos del botón. Seleccionamos el botón en la vista de diseño de la clase y nos vamos al **Property Inspector**.

Podemos observar que se nos abre una lista con las propiedades del componente tal como se muestra en la figura 5. La primera pestaña se trata de las propiedades y la segunda de los eventos. En este punto son los eventos lo que nos interesa.

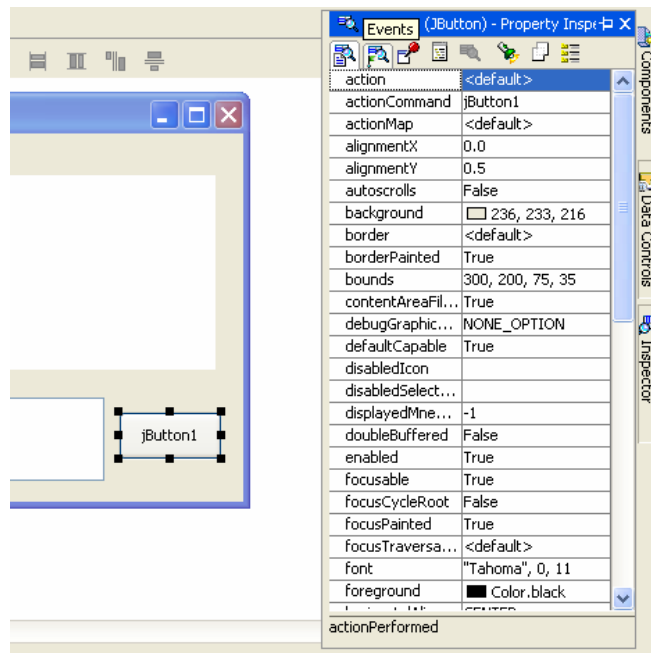


Figura 5

Seleccionamos eventos y clicamos en la casilla que aparece en blanco al lado de **Action Performed** y se nos abra la siguiente pantalla y aceptamos. Tal como podemos observar en la figura 6

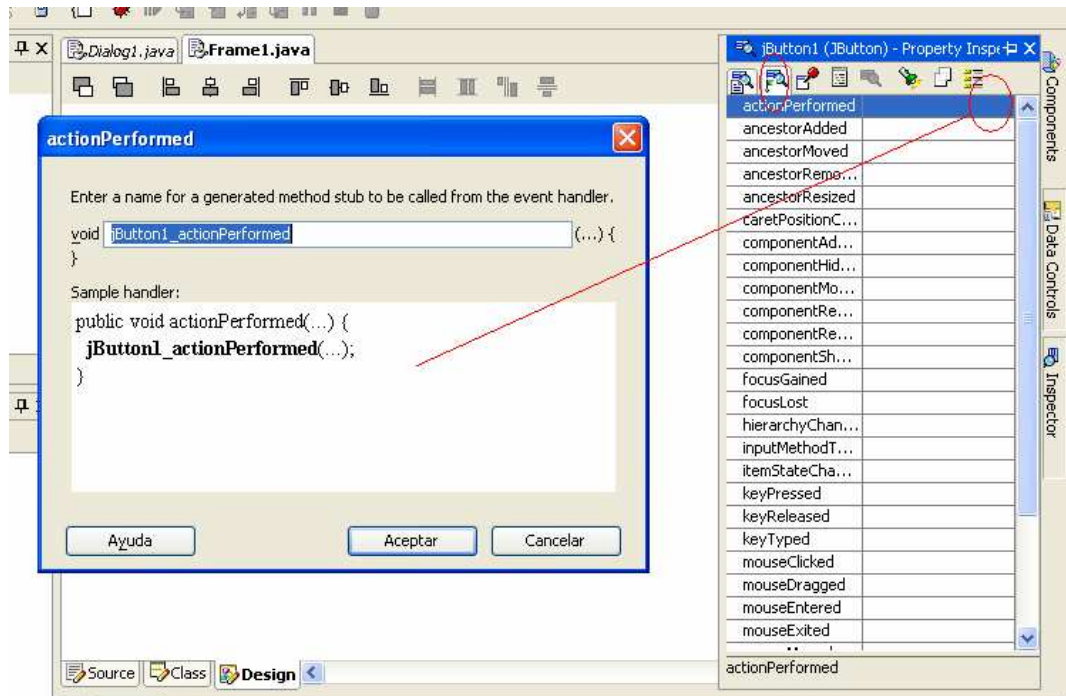


Figura 6

Esto lo haremos con todos los componentes que queramos controlar, de tal forma que después de hacerlo en nuestro textarea.

Nuestro código es el que se muestra en la figura 7:

Podemos observar que simplemente tenemos una serie de componentes gráficos y dos métodos que están vacíos que se encargaran de gestionar los eventos .Estos métodos están vacíos y serán implementados por nosotros mismos.

```
package mypackage;
import javax.swing.JFrame;
import java.awt.Dimension;
import javax.swing.JButton;
import java.awt.Rectangle;
import javax.swing.JCheckBox;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class Frame1 extends JFrame
{
    private JButton jButton1 = new JButton();
    private JTextArea jTextArea1 = new JTextArea();
    private JTextField jTextField1 = new JTextField();

    public Frame1()
    {
        try
        {
            jblnit();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }

    private void jblnit() throws Exception
    {
        this.getContentPane().setLayout(null);
        this.setSize(new Dimension(400, 300));
        jButton1.setText("jButton1");
        jButton1.setBounds(new Rectangle(300, 200, 75, 35));
        jButton1.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                jButton1_actionPerformed(e);
            }
        });
        jTextArea1.setText("jTextArea1");
        jTextArea1.setBounds(new Rectangle(15, 30, 355, 140));
        jTextField1.setText("jTextField1");
        jTextField1.setBounds(new Rectangle(15, 190, 275, 60));
        jTextField1.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                jTextField1_actionPerformed(e);
            }
        });
        this.getContentPane().add(jTextField1, null);
        this.getContentPane().add(jTextArea1, null);
        this.getContentPane().add(jButton1, null);
    }

    //metodo vacio que control evento de boton
    private void jButton1_actionPerformed(ActionEvent e)
    {
    }

    //metodo vacio que controla textfield
    private void jTextField1_actionPerformed(ActionEvent e)
    {
    }
}
```

Figura 7

Bueno, ya tenemos nuestra interfaz creada en swing. Espero que esto sirva de ayuda como iniciación a Jdeveloper y su editor grafico.

Destacar una vez más que el objetivo de este tutorial es crear una interfaz grafica con el editor grafico de Jdeveloper pero no explotar todas sus características y será el programador quien implemente los métodos para gestionar los eventos.

Soide

Cualquier Duda: [**soidemersolblog@yahoo.es**](mailto:soidemersolblog@yahoo.es)