

EJERCICIO GUIADO. JAVA: DISEÑO DE FORMULARIOS DESDE CÓDIGO

La Ventana de Diseño

La ventana de diseño es una gran herramienta que nos permite diseñar formularios de forma relativamente sencilla. Simplemente tenemos que añadir los componentes del formulario: botones, etiquetas, cuadros de textos, etc, y cambiar sus propiedades según nos interese.

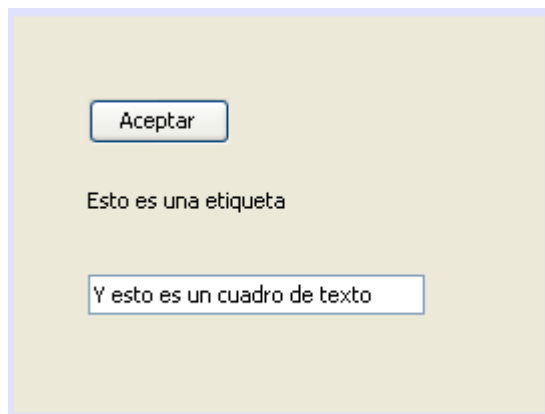
Todo esto tiene un código asociado que se genera de forma automática, y del que no nos hemos preocupado hasta el momento.

Por ejemplo, cada vez que añades un botón o una etiqueta, se generan automáticamente las instrucciones de código que permiten crear dicho botón o dicha etiqueta. También se genera el código que permite cambiar el texto del botón, o cambiar el texto de la etiqueta.

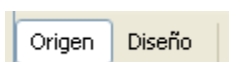
Un buen ejercicio para comprobar esto puede ser el siguiente:

Ejercicio guiado 1.

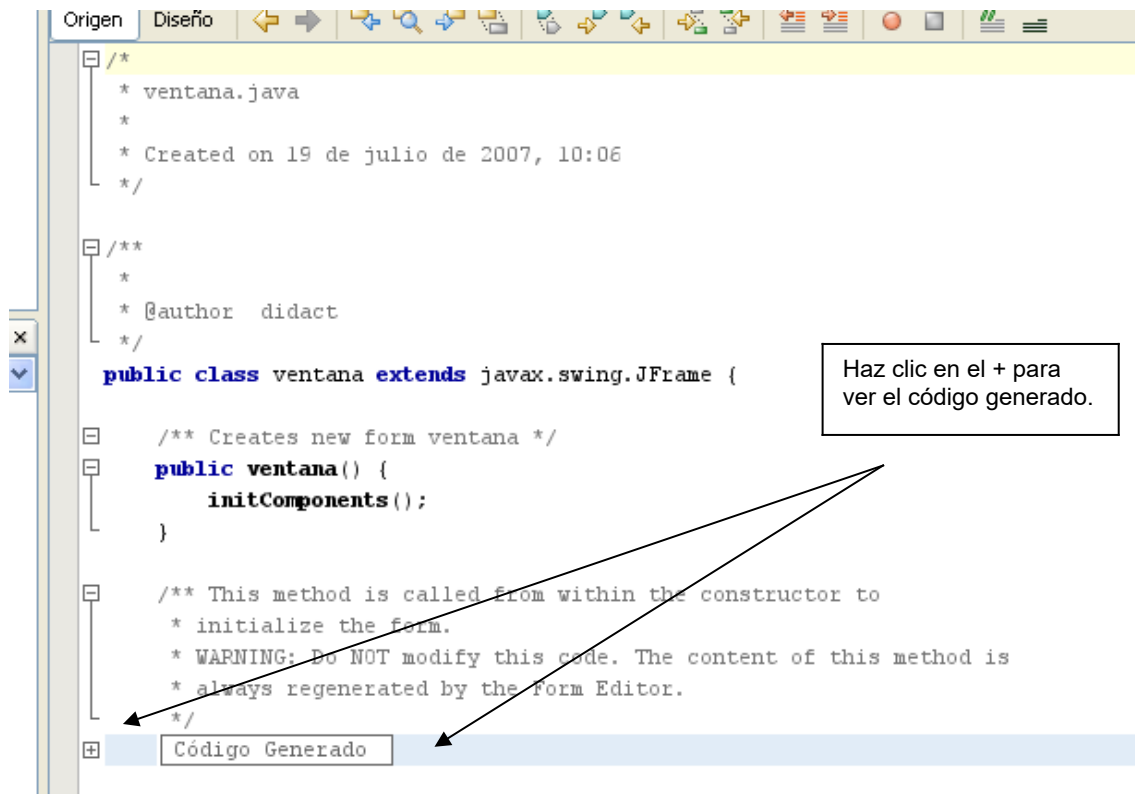
1. Crea un proyecto y añade en su ventana un botón (JButton), una etiqueta (JLabel) y un cuadro de texto (JTextField).
2. El nombre del botón será "btnBoton", el de la etiqueta "etiEtiqueta" y el del cuadro de texto "txtCuadroTexto".
3. A continuación asigna un texto a cada elemento. La ventana final podría tener un aspecto como el que sigue:



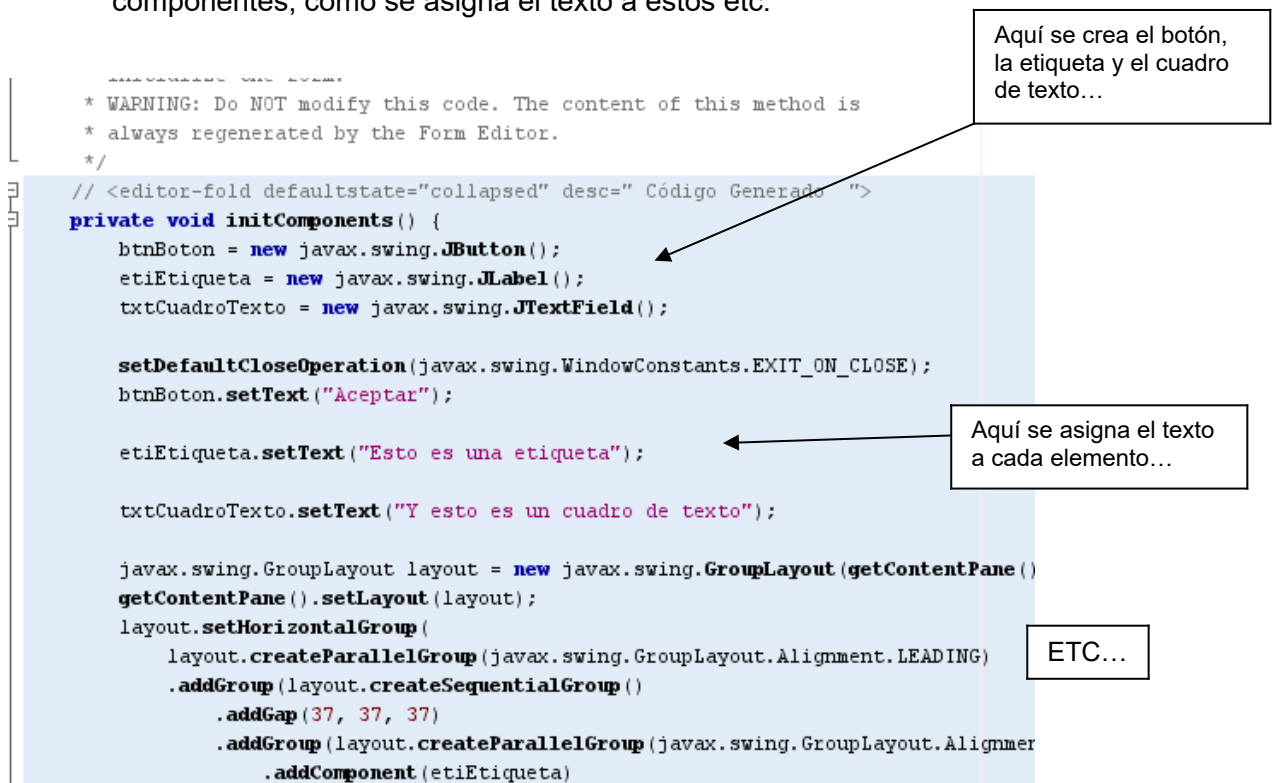
4. Todas estas operaciones que has realizado tienen asociado unas instrucciones de código que se han generado automáticamente. Para estudiar estas instrucciones, activa el botón *Origen* en la parte superior de la pantalla:



5. En el código, observarás una línea llamada *Código Generado*. Esta línea está señalada en color azul (al igual que todas las instrucciones generadas automáticamente) Observarás que tiene un + en la parte izquierda. Si haces clic sobre el signo + aparecerá el código generado automáticamente:



6. El código generado (señalado en azul) no puede ser modificado, sin embargo es interesante estudiarlo para entenderlo. En él puedes encontrar como se crean los componentes, como se asigna el texto a éstos etc.



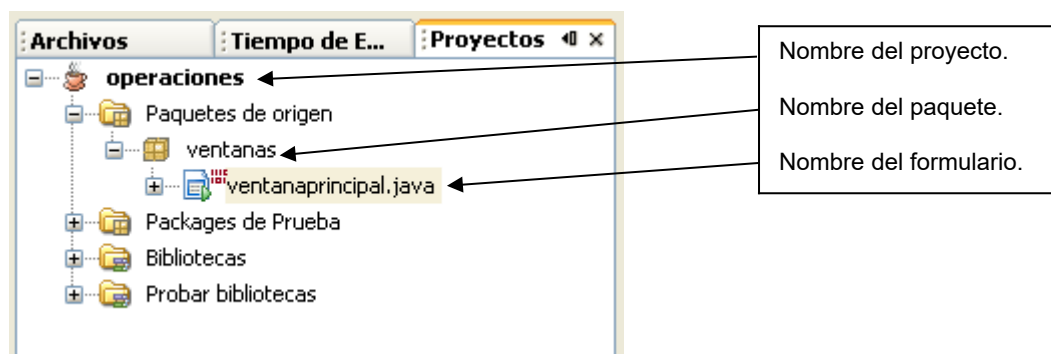
En esta explicación guiada, se diseñará el formulario entero directamente desde código, sin usar la ventana de diseño. De esta forma, se podrá entender mejor la generación de código del NetBeans y se tendrá más control sobre los componentes.

Ejercicio guiado 2

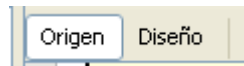
A. CREACIÓN DEL PROYECTO

1. Crear un nuevo proyecto con las siguientes características:
 - a. El nombre del proyecto será OPERACIONES.
 - b. El nombre del paquete será VENTANAS.
 - c. El nombre del formulario será VENTANAPRINCIPAL.

El aspecto que tendrá la ventana de proyectos debe ser el siguiente:



2. Una vez creado el proyecto, iremos directamente a la ventana de código, pulsando el botón *Origen*. No usaremos la ventana de diseño en este proyecto.



B. ESTRUCTURA DEL CÓDIGO (*estructura de una clase*)

3. Analizaremos el código del programa ahora. Empezaremos por arriba:

```
/*
 * ventanaprincipal.java
 *
 * Created on 19 de julio de 2007, 10:54
 */

package ventanas;

/**
 *
 * @author didact
 */
```

En la parte superior del código verás un comentario.

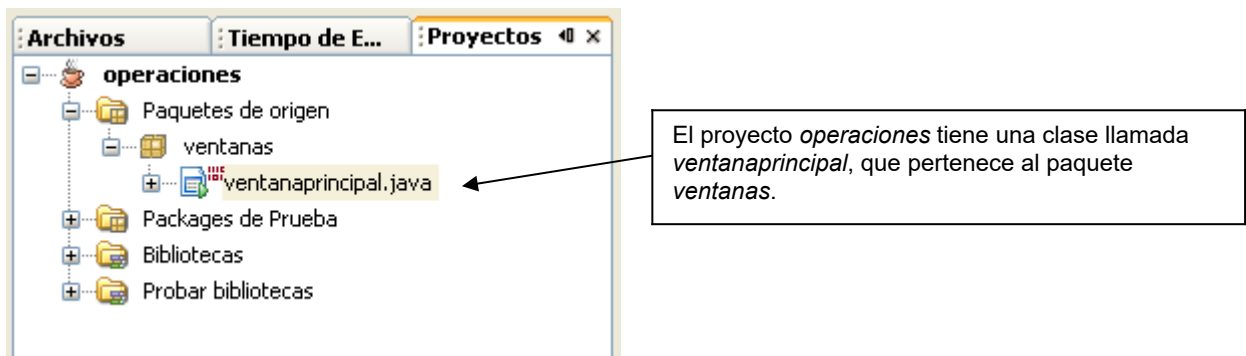
En este comentario aparece el nombre del fichero que contiene el código, que como puedes observar tiene el mismo que el del JFrame.

También aparece la fecha de creación del fichero.

Observarás también una instrucción que indica que este fichero (*ventanaprincipal.java*) pertenece al paquete *ventanas*.

También verás otro comentario donde aparece el nombre del autor. Este comentario se puede cambiar para que aparezca tu nombre.

4. Un proyecto en java está formado por módulos denominados *CLASES*. El proyecto que acabamos de crear tiene un solo módulo: el JFrame que define la ventana principal, llamado precisamente *ventanaprincipal*. Por tanto, se puede decir que nuestro proyecto tiene una clase llamada *ventanaprincipal*.



5. Para programar una clase se sigue la siguiente sintaxis:

```
public class nombredelaclase {  
  
    ...programación de la clase...  
}
```

6. Esto lo puedes observar claramente en el código de tu proyecto:

```

public class ventanaprincipal extends javax.swing.JFrame {

    /** Creates new form ventanaprincipal */
    public ventanaprincipal() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // Código Generado

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new ventanaprincipal().setVisible(true);
            }
        });
    }

    // Declaración de varibales -no modificar
    // Fin de declaración de variables
}

```

Programación de la clase *ventanaprincipal*

7. Dentro de la clase, tenemos una serie de funciones y procedimientos llamados **MÉTODOS**. Es en los métodos donde se programan los eventos que pueden suceder sobre el formulario.
8. Hay un método especial muy importante denominado **CONSTRUCTOR**. Dentro de este método se programa el diseño de la ventana y las características iniciales de la ventana.

El constructor se puede distinguir fácilmente porque tiene el mismo nombre que la clase. En nuestro ejemplo, *ventanaprincipal*.

```

/** Creates new form ventanaprincipal */
public ventanaprincipal() {
    initComponents();
}

```

Observarás que el constructor contiene una llamada a un método llamado *initComponents*. El método *initComponents* define características iniciales de la ventana, y es código generado automáticamente. Si quieres verlo, solo tienes que desplegar la parte de *Código Generado*:

```
// <editor-fold defaultstate="collapsed" desc=" Código Generado ">
private void initComponents() {

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(0, 400, Short.MAX_VALUE)
            )
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(0, 300, Short.MAX_VALUE)
            )
    );
    pack();
}
// </editor-fold>
```

Clic aquí para ocultar el código generado

El método *initComponents* al que hace referencia el constructor es un método generado automáticamente, y no puede ser modificado.

Puedes verlo dentro de la sección de Código Generado.

Este método define algunas características iniciales de la ventana.

El contenido del método *initComponents* no resulta interesante para el ejercicio que estamos realizando, así que puedes ocultar de nuevo el *Código Generado*, haciendo clic en el símbolo menos -

9. Otro método importante que hay que mencionar es el método *main*. Este método es el que se ejecuta cuando arranca el programa. Se puede decir que el programa parte de aquí.

La tarea que desempeña este método es la de generar la ventana principal y mostrarla en pantalla.

El método *main*, es similar a la función *main* del lenguaje C. En el caso del java, el NetBeans genera automáticamente el método *main*, de forma que no tendremos que hacer cambios en él.

C. DISEÑO DE LA VENTANA DESDE CÓDIGO

Vamos ahora a diseñar la ventana de nuestro programa (sin usar la ventana de diseño, solo a partir de código). Al final de esta explicación, la ventana del proyecto tendrá el siguiente aspecto:

Para conseguir esto, sigue los siguientes pasos:

10. Para realizar el diseño de la ventana, crearemos un método propio al que llamaremos por ejemplo *CreacionVentana*. Un buen sitio para colocar este método puede ser debajo del constructor:

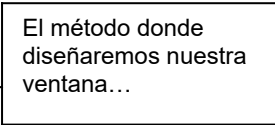
```

public class ventanaprincipal extends javax.swing.JFrame {

    /** Creates new form ventanaprincipal */
    public ventanaprincipal() {
        initComponents();
    }

    public void CreacionVentana() {

```



```

    }

```

11. Lo primero que se va a hacer es definir características propias de la ventana, como por ejemplo su título, tamaño, etc. Recuerda que la forma de hacer referencia a la propia ventana es a través de la palabra reservada *this*.

Así pues, programa lo siguiente dentro del método *CreacionVentana*:

```

public void CreacionVentana() {

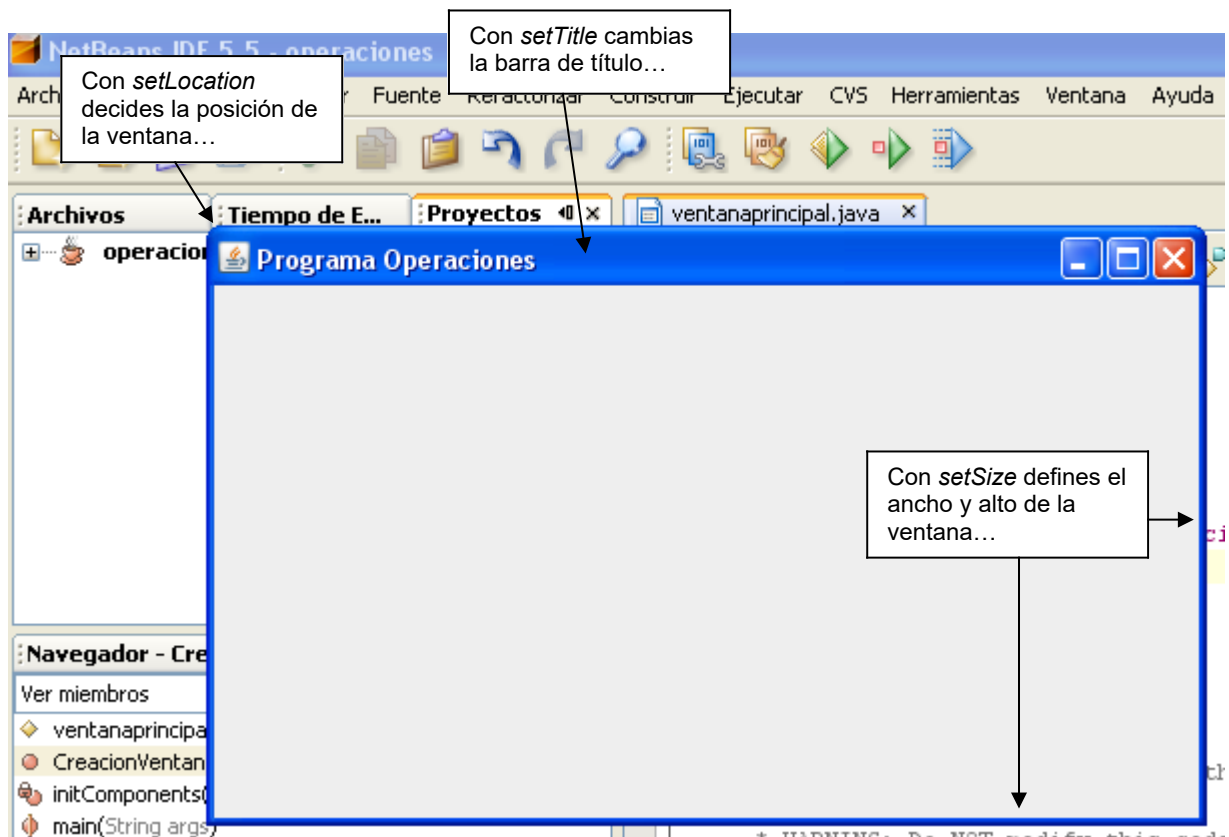
    this.setTitle("Programa Operaciones");
    this.setSize(500,300);
    this.setLocation(100,100);

}

```

Este código hace lo siguiente:

- Usa el método *setTitle* de la ventana para definir el texto de la barra de título.
- Usa el método *setSize* de la ventana para definir un tamaño de 500x300.
- Usa el método *setLocation* de la ventana para definir la posición de la ventana en la posición (100,100) de la pantalla.



12. Para que esto tenga efecto, es necesario que el método *CrearVentana* sea llamado desde el constructor, al igual que sucede con el método *initComponents* :

```
public class ventanaprincipal extends javax.swing.JFrame {

    /** Creates new form ventanaprincipal */
    public ventanaprincipal() {
        initComponents();
        CreacionVentana();
    }

    public void CreacionVentana() {

        this.setTitle("Programa Operaciones");
        this.setSize(500,300);
        this.setLocation(100,100);

    }
}
```

13. Ya puedes ejecutar el programa. En el momento en que se ejecuta el programa, el constructor llama al método *CreacionVentana* y se definen las características programadas.

14. Ahora vamos a empezar a añadir componentes a la ventana. Concretamente, empezaremos creando la etiqueta que contiene el texto "Número 1:" y luego definiremos algunas características propias para finalmente situarla en la ventana.

Al comienzo de la clase, después del *public class*, donde se suelen definir las variables globales, añade el siguiente código:

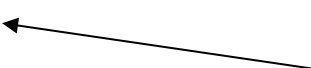
```
public class ventanaprincipal extends javax.swing.JFrame {  
    JLabel etiNum1;
```

Este código declara una variable de tipo JLabel (etiqueta) y le da el nombre *etiNum1*.

15. Al escribir este código, aparecerá un error en la línea, ya que será necesario importar la librería donde se encuentra el tipo de objeto *JLabel*. Podrás usar la bombilla del NetBeans para agregar el *import* correspondiente automáticamente.

16. Los *imports* aparecerán en la parte superior del código. Puedes observarlo:

```
package ventanas;  
  
import javax.swing.JLabel;  
  
/**  
 *  
 * @author didact  
 */  
  
public class ventanaprincipal extends javax.swing.J
```



Importación de la librería para el objeto JLabel, que define las etiquetas.

(Agregada automáticamente al pulsar la bombilla del NetBeans)

17. Ahora agrega este código a tu método *CreacionVentana*:

```
public void CreacionVentana() {  
  
    this.setTitle("Programa Operaciones");  
    this.setSize(500,300);  
    this.setLocation(100,100);  
  
    etiNum1 = new JLabel();  
    etiNum1.setText("Número 1:");  
    etiNum1.setBounds(10,10,100,20);  
    this.getContentPane().add(etiNum1);  
}
```

18. La explicación del código es la siguiente:

- La primera línea *construye* la etiqueta llamada *etiNum1* que fue declarada antes.

```
etiNum1 = new JLabel();
```
- Generalizando: para crear un objeto llamado xxx del tipo *tipoobjeto*, se tendrá que hacer lo siguiente:

- En la zona de las variables globales se declarará la variable del objeto de la siguiente forma:

```
tipoobjeto xxx;
```

- Luego, dentro de la clase, se construye el objeto siguiendo la siguiente sintaxis:

```
xxx = new tipoobjeto();
```

- En el ejemplo, la etiqueta creada, cuyo nombre es *etiNum1*, tiene que contener el texto "Número 1:". Esto lo hacemos con el típico método *setText*:

```
etiNum1.setText("Número 1: ");
```

- Hay que definir la posición que tendrá el objeto y el tamaño. Para ello, se usará el método *setBounds*. Este método recibe la posición (x, y) del objeto y el ancho y el alto. En el ejemplo, la etiqueta está situada en la posición (10,10) de la ventana y tiene un ancho de 100 y un alto de 20:

```
etiNum1.setBounds(10,10,100,20);
```

- Finalmente, hay que colocar la etiqueta en el panel de contenidos de la ventana. Para ello, se usa el método *add* del panel de contenidos. Recuerda que para obtener este panel se usa el método *getContentPane*:

```
this.getContentPane().add(etiNum1);
```

19. Es buena idea que ejecutes el programa para ver el aspecto de la ventana ahora.

20. Hemos añadido una etiqueta. Ahora añadiremos la siguiente. Para ello, primero tienes que declarar la etiqueta en la zona de las variables globales:

```
L */
public class ventanaprincipal extends javax.swing.JFrame {

    JLabel etiNum1;
    JLabel etiNum2; ←
```

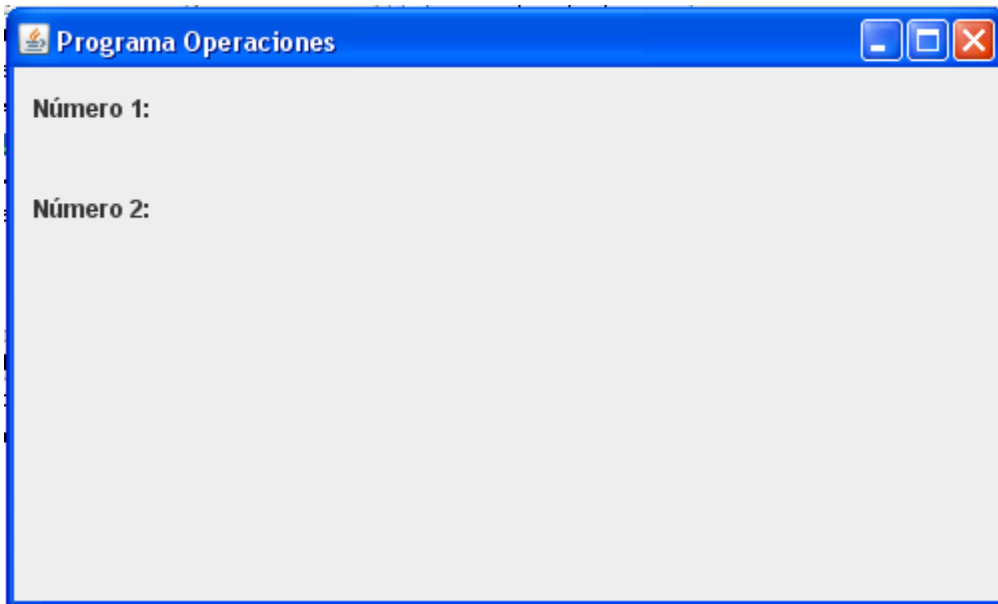
21. Luego, en el método *ConstruirVentana*, añadiremos el siguiente código:

```
etiNum1 = new JLabel();
etiNum1.setText("Número 1:");
etiNum1.setBounds(10,10,100,20);
this.getContentPane().add(etiNum1);
```

```
etiNum2 = new JLabel();
etiNum2.setText("Número 2:");
etiNum2.setBounds(10,60,100,20);
this.getContentPane().add(etiNum2);
```

22. Este código es prácticamente igual al anterior. En él se construye la etiqueta declarada antes con el nombre *etiNum2*. Luego se le asigna el texto "Número 2:", luego se usa el método *setBounds* para situarla en la posición (10, 60) y asignarle un ancho de 100 y un alto de 20. Finalmente se añade la etiqueta al panel de contenidos de la ventana.

23. Ejecuta el programa y observa el aspecto de la ventana ahora:



24. Sigamos añadiendo elementos. Ahora añadiremos un cuadro de texto llamado *txtNum1*. Primero declara la variable del cuadro de diálogo en la zona de variables globales:

```
public class ventanaprincipal extends javax.swing.JFrame {  
  
    JLabel etiNum1;  
    JLabel etiNum2;  
    JTextField txtNum1; ←
```

Tendrás que agregar el import correspondiente para el tipo de objeto *JTextField*.

25. Luego añade el siguiente código en el método *CreacionVentana*:

```
txtNum1 = new JTextField();  
txtNum1.setText("0");  
txtNum1.setBounds(110,10,100,20);  
this.getContentPane().add(txtNum1);
```

26. Este código realiza lo siguiente:

- La primera línea construye el cuadro de texto llamado *txtNum1*. Los *JTextField* son objetos del tipo cuadro de texto.

- Esta línea se subrayará en rojo ya que será necesario añadir el *import* para la librería de los JTextField.
- La siguiente línea introduce la cadena "0" dentro del cuadro de texto.
- La siguiente línea asigna la posición y el tamaño a través del método *setBounds*.
- Y finalmente, la última línea añade el cuadro de texto al panel de contenidos de la ventana.

27. Ejecuta el programa y comprueba el funcionamiento de este código.

28. Ahora se añadirá un nuevo cuadro de texto llamado *txtNum2*. El código es similar. Declara la variable global:

```
public class ventanaprincipal extends javax.swing.JFrame {

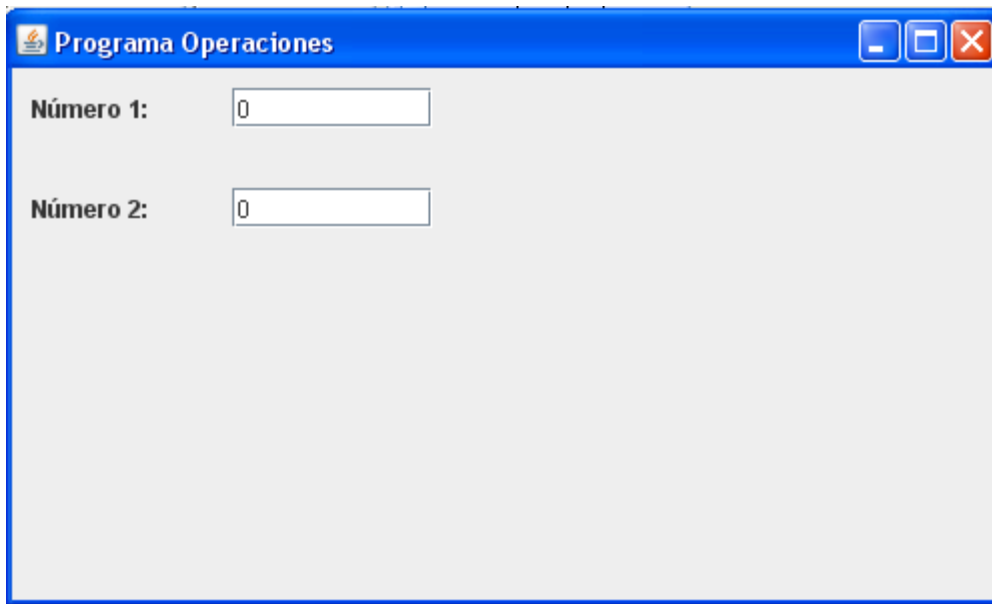
    JLabel etiNum1;
    JLabel etiNum2;
    JTextField txtNum1;
    JTextField txtNum2; ←
```

29. Y ahora añade el código que construye el objeto en el método *ConstruirVentana*. Este código es similar al del cuadro de texto anterior. Solo cambia el nombre del objeto y la posición donde se colocará:

```
txtNum1 = new JTextField();
txtNum1.setText("0");
txtNum1.setBounds(110,10,100,20);
this.getContentPane().add(txtNum1);
```

```
txtNum2 = new JTextField();
txtNum2.setText("0");
txtNum2.setBounds(110,60,100,20);
this.getContentPane().add(txtNum2);
```

30. Ejecuta el programa y comprueba como va el diseño de la ventana:



31. Ahora le toca el turno a los botones. Primero declararemos una variable del tipo `JButton` en la zona de variables globales (será necesario añadir el *import* correspondiente a la clase `JButton` a través de la bombilla). El botón se llamará *btnSumar*:

```
public class ventanaprincipal extends javax.swing.JFrame {  
  
    JLabel etiNum1;  
    JLabel etiNum2;  
    JTextField txtNum1;  
    JTextField txtNum2;  
    JButton btnSumar; ←
```

32. Y a continuación construiremos y le daremos forma al botón, añadiendo el siguiente código en el método *ConstruirVentana*:

```
txtNum2 = new JTextField();  
txtNum2.setText("0");  
txtNum2.setBounds(110,60,100,20);  
this.getContentPane().add(txtNum2);  
  
btnSumar = new JButton();  
btnSumar.setText("Sumar");  
btnSumar.setBounds(10,120,100,20);  
this.getContentPane().add(btnSumar);  
|
```

33. La explicación del código anterior es similar a los anteriores:

- La primera línea construye el botón `btnSumar`.
- A continuación se asigna el texto "Sumar" al botón.
- Luego se asigna la posición y el tamaño del botón con el método *setBounds*.
- Finalmente se añade el botón al panel de contenido de la ventana.

34. Ejecuta el programa y observa el aspecto de la ventana.

35. Ahora se añadirá otro botón llamado btnRestar. Declara la variable JButton:

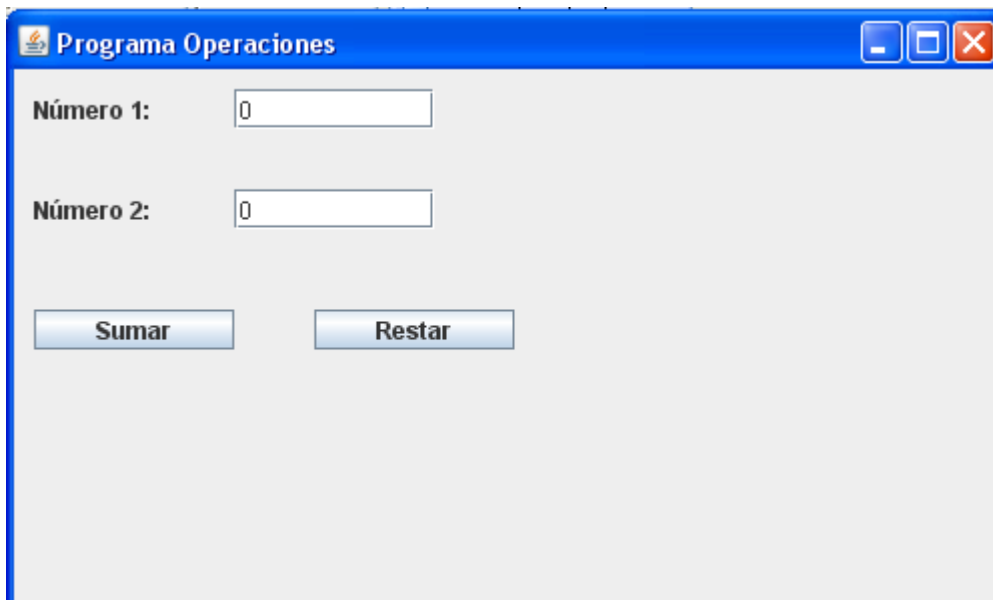
```
public class ventanaprincipal extends javax.swing.JFrame {  
  
    JLabel etiNum1;  
    JLabel etiNum2;  
    JTextField txtNum1;  
    JTextField txtNum2;  
    JButton btnSumar;  
    JButton btnRestar; ←
```

36. Tienes que escribir el siguiente código:

```
btnSumar = new JButton();  
btnSumar.setText("Sumar");  
btnSumar.setBounds(10,120,100,20);  
this.getContentPane().add(btnSumar);  
  
btnRestar = new JButton();  
btnRestar.setText("Restar");  
btnRestar.setBounds(150,120,100,20);  
this.getContentPane().add(btnRestar);
```

37. El código es prácticamente igual que el del primer botón. Solo cambia el nombre del botón, el texto y la posición y tamaño del botón.

38. Ejecuta el programa para ver como queda la ventana.



CONCLUSIÓN

La ventana de diseño de NetBeans es una herramienta que nos permite diseñar las ventanas sin tener que programar código.

Es posible prescindir de la ventana de diseño y realizar todo el diseño de la ventana programando.

El diseño de la ventana se tiene que hacer en el constructor de la clase.

Los componentes de una ventana son variables globales cuyo tipo se corresponde a distintas clases de objetos: JLabel, JTextField, JButton, etc...

Para construir un objeto hay que declararlo como una variable global:

TipoObjeto NombreObjeto;

Y luego, dentro de la clase, construirlo, usando la siguiente sintaxis:

NombreObjeto = new TipoObjeto();

Luego se pueden usar distintos métodos del objeto para trabajar con él.