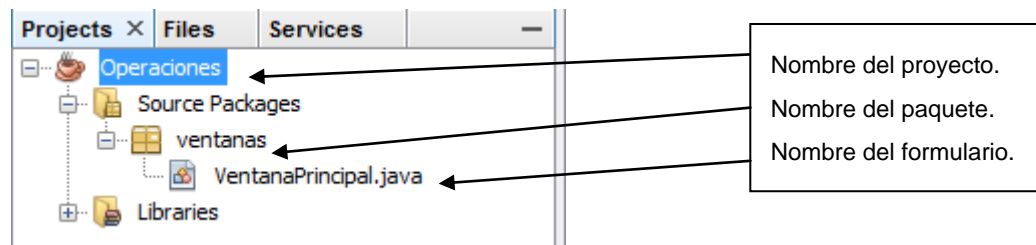


# EJERCICIOS 1

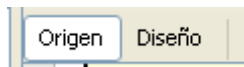
## Ejercicio guiado

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 de la clase que contendrá el 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*.



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

The screenshot shows the code editor with the following content:

```
/*  
 * ventanaprincipal.java  
 *  
 * Created on 19 de julio de 2007, 10:54  
 */  
  
package ventanas;  
  
/**  
 *  
 * @author didact  
 */
```

Annotations on the right side of the image:

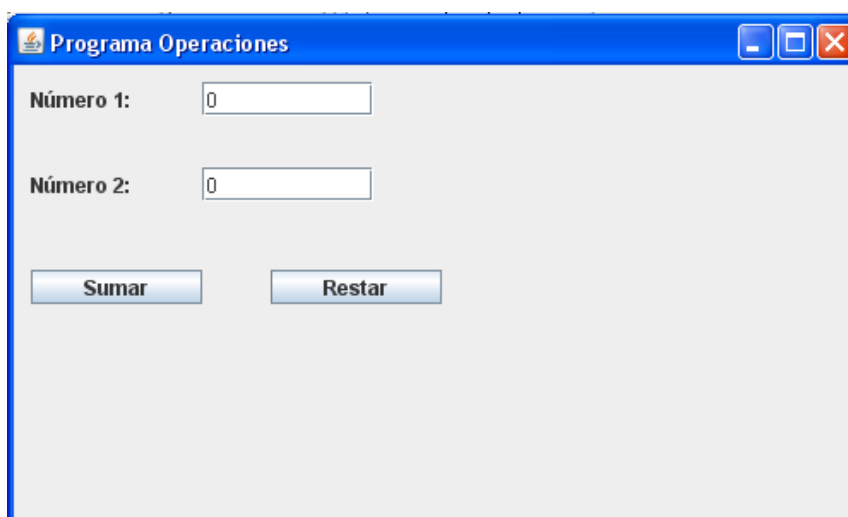
- En la parte superior del código verás un comentario.
- En este comentario aparece el nombre del fichero que contiene el código.
- También aparece la fecha de creación del fichero.
- Observarás también una instrucción que indica que este fichero (*ventanaprincipal.java*) pertenece al paquete *ventanas*.
- También verás otro comentario donde aparece el nombre del autor. Este comentario se puede cambiar para que aparezca tu nombre.

5. Dentro de la clase, en el constructor, se programa el diseño de la ventana y las características iniciales de la ventana.

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.

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

6. Comprueba cómo también se ha creado el método main, que es el que se ejecutará al arrancar el programa.
7. Vamos ahora a diseñar la ventana de nuestro programa. Al final de esta explicación, la ventana del proyecto tendrá el siguiente aspecto:



8. 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() {
```

El método donde  
diseñaremos nuestra  
ventana...

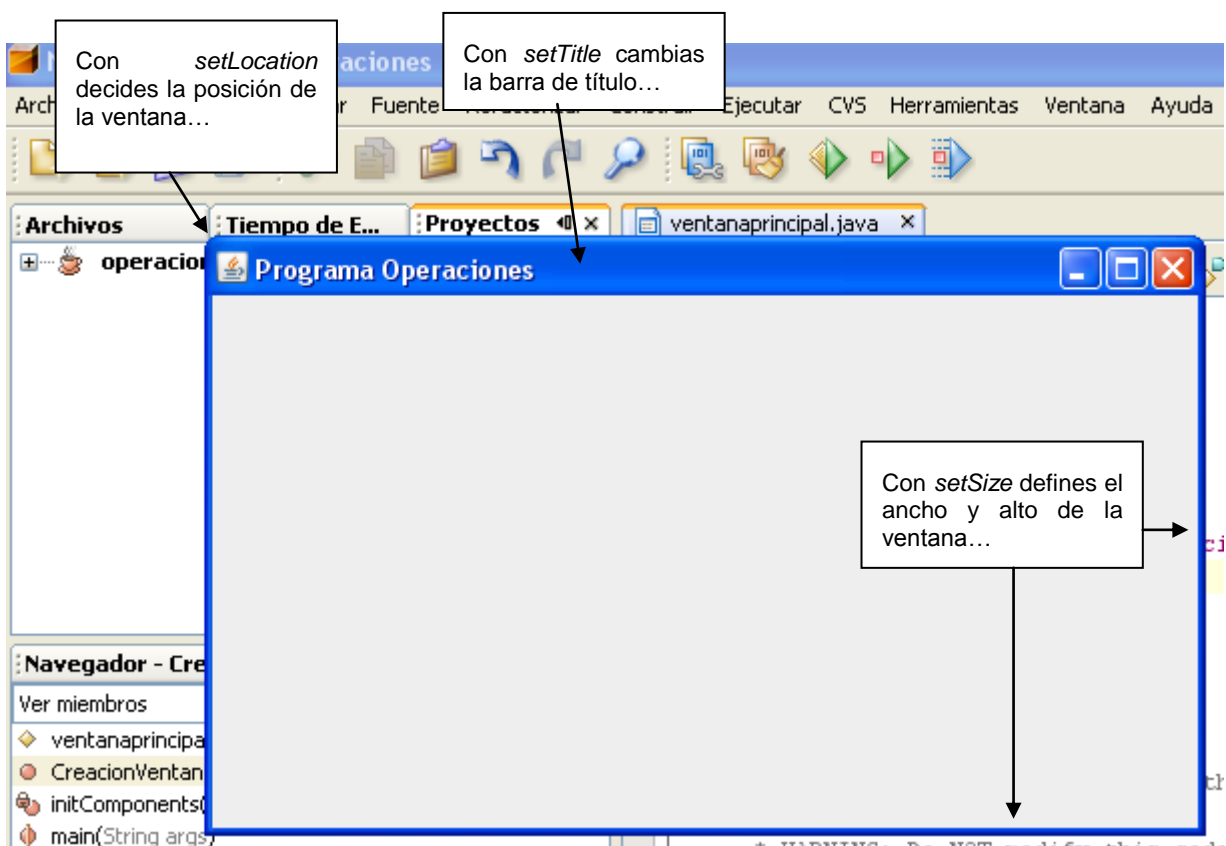
9. Lo primero que se va a hacer es definir características propias de la ventana, como por ejemplo su título, tamaño, etc. Recuerda que la forma de hacer referencia a la propia ventana es a través de la palabra reservada *this*.

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

```
public void CreacionVentana() {  
  
    this.setTitle("Programa Operaciones");  
    this.setSize(500,300);  
    this.setLocation(100,100);  
  
}
```

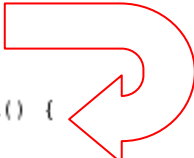
Este código hace lo siguiente:

- 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.



10. Para que esto tenga efecto, es necesario que el método *CrearVentana* sea llamado desde el constructor:

```
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);  
  
    }  
}
```

A red arrow originates from the `CreacionVentana();` line inside the `ventanaprincipal()` constructor and points to the `public void CreacionVentana()` method definition below it.

11. 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.


12. 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*. 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.

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

A black arrow points from the `import javax.swing.JLabel;` line to the text box on the right.

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

(Agregada automáticamente al pulsar la bombilla del NetBeans)

13. 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);  
}
```

14. La explicación del código es la siguiente:

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

**etiNum1 = new JLabel();**

- 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);**

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

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

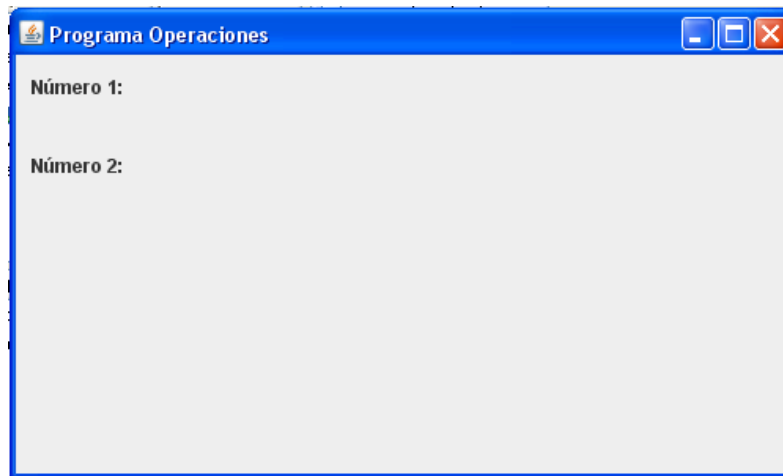
```
L */  
public class ventanaprincipal extends javax.swing.JFrame {  
  
    JLabel etiNum1;  
    JLabel etiNum2;  
}
```

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

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

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.

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



19. 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`.

20. 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);
```

21. Este código realiza lo siguiente:

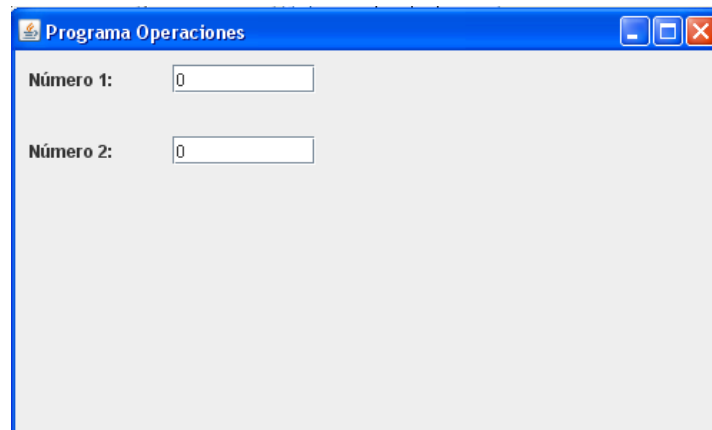
- La primera línea construye el cuadro de texto llamado *txtNum1*. Los `JTextField` son objetos del tipo cuadro de texto.
- La siguiente línea introduce la cadena "0" dentro del cuadro de texto.
- La siguiente línea asigna la posición y el tamaño a través del método *setBounds*.
- Y finalmente, la última línea añade el cuadro de texto al panel de contenidos de la ventana.

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

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

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

24. Ejecuta el programa y comprueba cómo va el diseño de la ventana:



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; |
```

25. 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);
```

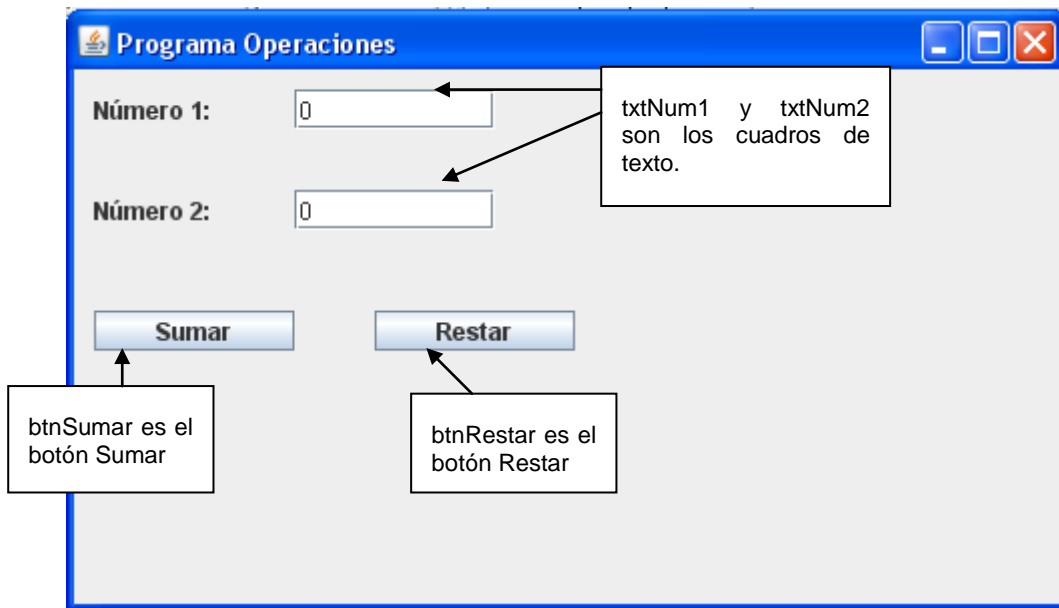
26. La explicación del código anterior es similar a los anteriores:

- 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.

27. Ahora se añadirá otro botón llamado `btnRestar`. Declara la variable `JButton` correspondiente y añade el siguiente código, al igual que hiciste con el botón de sumar (solo cambia el nombre del botón, el texto y la posición y tamaño del botón):

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

28. Ejecuta el programa para ver cómo queda la ventana.



29. A continuación programaremos la pulsación del botón **btnSumar** para que aparezca un *JOptionPane* con la suma calculada. Luego haremos lo mismo con el botón **btnRestar**.
30. Primero asignaremos al botón `btnSumar` un escuchador de eventos del tipo **actionPerformed**. Esto lo haremos al final del método *CreacionVentana*. El código que debe añadir para el evento `actionPerformed` del botón `btnSumar` es el siguiente:

```
btnSumar.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        btnSumarActionPerformed(evt);  
    }  
});
```

*(Recuerda que el código es siempre igual, solo hay que indicar el nombre del botón y el nombre del procedimiento al que se llama)*

31. Será necesario añadir un import para el objeto oyente `ActionListener` y para la clase `ActionEvent`.



32. Ahora que se ha definido un evento actionPerformed para el botón btnSumar, será necesario programarlo. Esto se hace creando el procedimiento al que llama el código del actionPerformed. A este procedimiento le hemos dado de nombre btnSumarActionPerformed.
33. Así pues, añade el siguiente procedimiento a la clase. ( el siguiente código está fuera del método CreacionVentana):

Este es el final del método CreacionVentana ...

```
btnRestar.setText("Restar");
btnRestar.setBounds(150,120,100,20);
this.getContentPane().add(btnRestar);

btnSumar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        btnSumarActionPerformed(evt);
    }
});
```

Hay que programar el método al que llama el evento...

```
public void btnSumarActionPerformed(ActionEvent evt) {

}
```

Este es el código que tienes que añadir

34. Es precisamente en este nuevo procedimiento que hemos creado donde se programa el evento actionPerformed del botón btnSumar. Lo que se pretende que haga el programa es que aparezca un JOptionPane con la suma de los números introducidos. Para ello, añade el siguiente código:

```
public void btnSumarActionPerformed(ActionEvent evt) {
    double a,b,s;

    a=Double.parseDouble(txtNum1.getText());
    b=Double.parseDouble(txtNum2.getText());
    s=a+b;
    JOptionPane.showMessageDialog(null,"La suma es "+s);
}
```

Programa el código de lo que tiene que hacer el evento.

35. Bien, hemos programado la pulsación del botón Sumar. Hemos seguido dos pasos:
- Primero hemos creado el evento actionPerformed y se lo hemos asignado al botón. Esto lo hemos hecho dentro del método CreacionVentana. Este evento hace una llamada al procedimiento donde se programará la respuesta al evento.
  - Luego hemos creado ese procedimiento y hemos programado en él el código correspondiente a la pulsación del botón.

36. Ejecuta el programa y comprueba el funcionamiento del botón Sumar.

37. Ahora programaremos el botón **btnRestar**. Primero se tendrá que definir el evento `actionPerformed` y asignárselo al botón. Esto se hace añadiendo el código de creación del evento al final del método `CreacionVentana`.

```
btnRestar.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        btnRestarActionPerformed(evt);  
    }  
});
```

*(Observa que es idéntico al código de asignación del evento del botón `btnSumar`, solo cambia el nombre del botón y el nombre del procedimiento al que se llama)*

38. Y ahora, hay que programar el procedimiento al que llama el evento **actionPerformed**. Es aquí donde se programa la respuesta al evento. En nuestro ejemplo, queremos que al pulsar el botón Restar se resten los números introducidos.

```
public void btnRestarActionPerformed(ActionEvent evt) {  
    double a,b,r;  
  
    a=Double.parseDouble(txtNum1.getText());  
    b=Double.parseDouble(txtNum2.getText());  
    r=a-b;  
    JOptionPane.showMessageDialog(null,"La resta es "+r);  
}
```

39. Ejecuta el programa y comprueba el funcionamiento del botón Restar.

40. Resumiendo una vez más. **Para programar un evento sobre un componente, primero hay que enlazar el oyente del evento con el componente, y luego programar el método al que llama el evento.** Observa el código que hemos programado:

```
btnSumar.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        btnSumarActionPerformed(evt);  
    }  
});
```

Asignación de un *actionPerformed* al botón btnSumar.

(Se hace una llamada al método *btnSumarActionPerformed*)

```
btnRestar.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        btnRestarActionPerformed(evt);  
    }  
});
```

Asignación de un *actionPerformed* al botón btnRestar.

(Se hace una llamada al método *btnRestarActionPerformed*)

```
public void btnSumarActionPerformed(ActionEvent evt) {  
    double a,b,s;  
  
    a=Double.parseDouble(txtNum1.getText());  
    b=Double.parseDouble(txtNum2.getText());  
    s=a+b;  
    JOptionPane.showMessageDialog(null,"La suma es "+s);  
}
```

Programación del método *btnSumarActionPerformed*

(respuesta a la pulsación del botón btnSumar)

```
public void btnRestarActionPerformed(ActionEvent evt) {  
    double a,b,r;  
  
    a=Double.parseDouble(txtNum1.getText());  
    b=Double.parseDouble(txtNum2.getText());  
    r=a-b;  
    JOptionPane.showMessageDialog(null,"La resta es "+r);  
}
```

Programación del método *btnRestarActionPerformed*

(respuesta a la pulsación del botón btnRestar)

## CONCLUSIONES

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.

Cada evento pertenece a un objeto oyente, y es el oyente el que se asigna al componente de la ventana que se quiere programar.

El evento *actionPerformed* pertenece al oyente *ActionListener*.

Para enlazar el oyente *ActionListener* a un componente XXX hay que usar el siguiente código:

```
XXX.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        XXXActionPerformed(evt);  
    }  
});
```

El enlace de un evento a un componente se hace en el constructor. Esta asignación hace una llamada a un procedimiento XXXActionPerformed, (donde XXX representa el nombre del componente que se programa) Es en este procedimiento donde realmente se programa el evento.