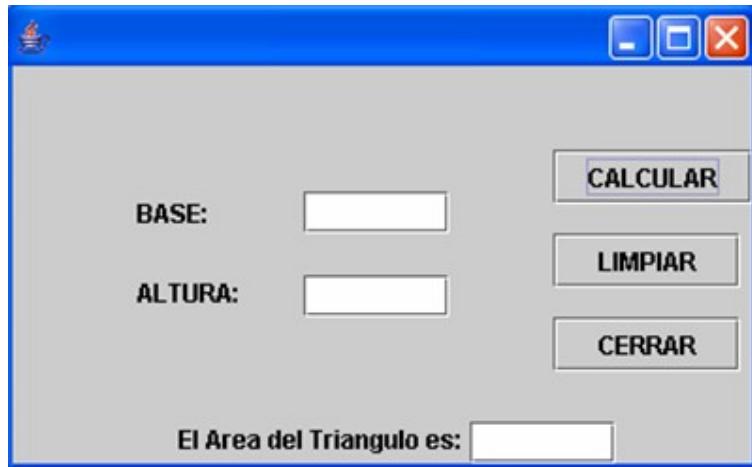


USO DE LOS OBJETOS JLABEL, JTEXTFIELD Y JBUTTON



Una aplicación sin usar Formulario

A continuación vamos a desarrollar una aplicación sencilla que permita calcular el área del triángulo dado los valores de la base y la altura. No se utilizará formulario, pero sí las clases del paquete swing para ingresos y salida de datos.

Solución:

Estando en el entorno de desarrollo NetBeans seleccionamos la opción del menú y luego la opción **New Project**.

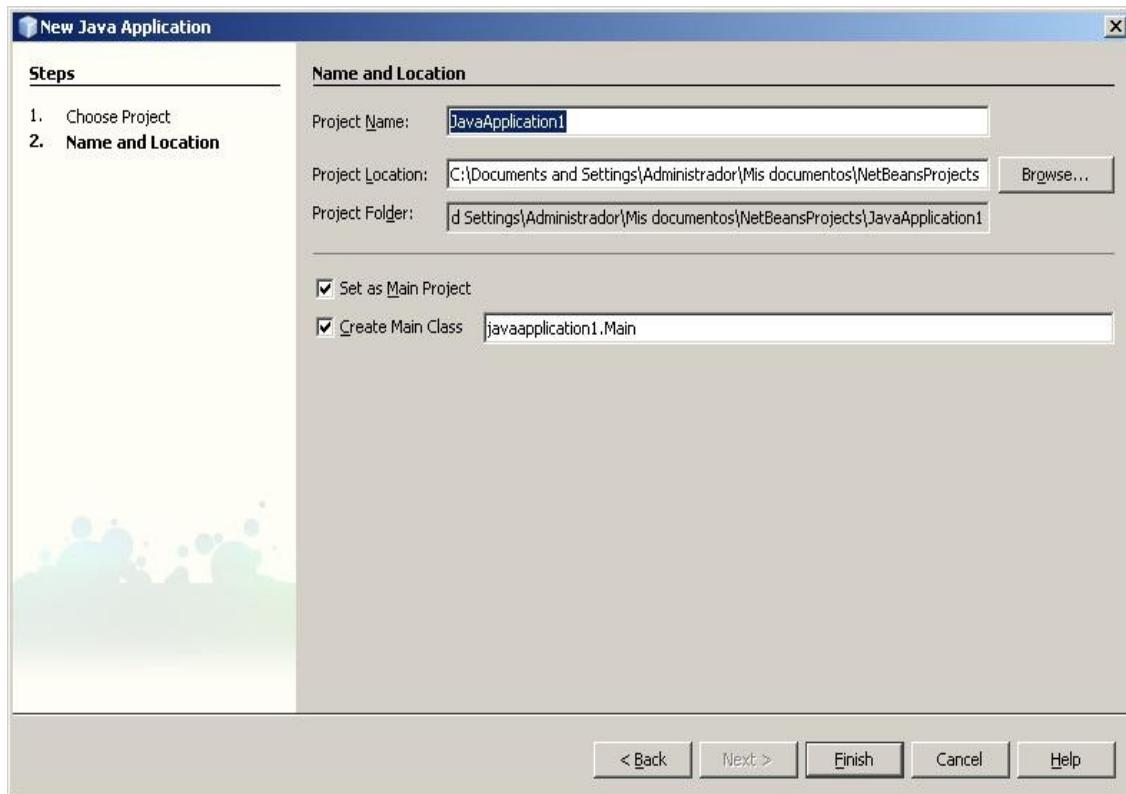


Al momento de seleccionar New Project se visualiza la siguiente ventana:



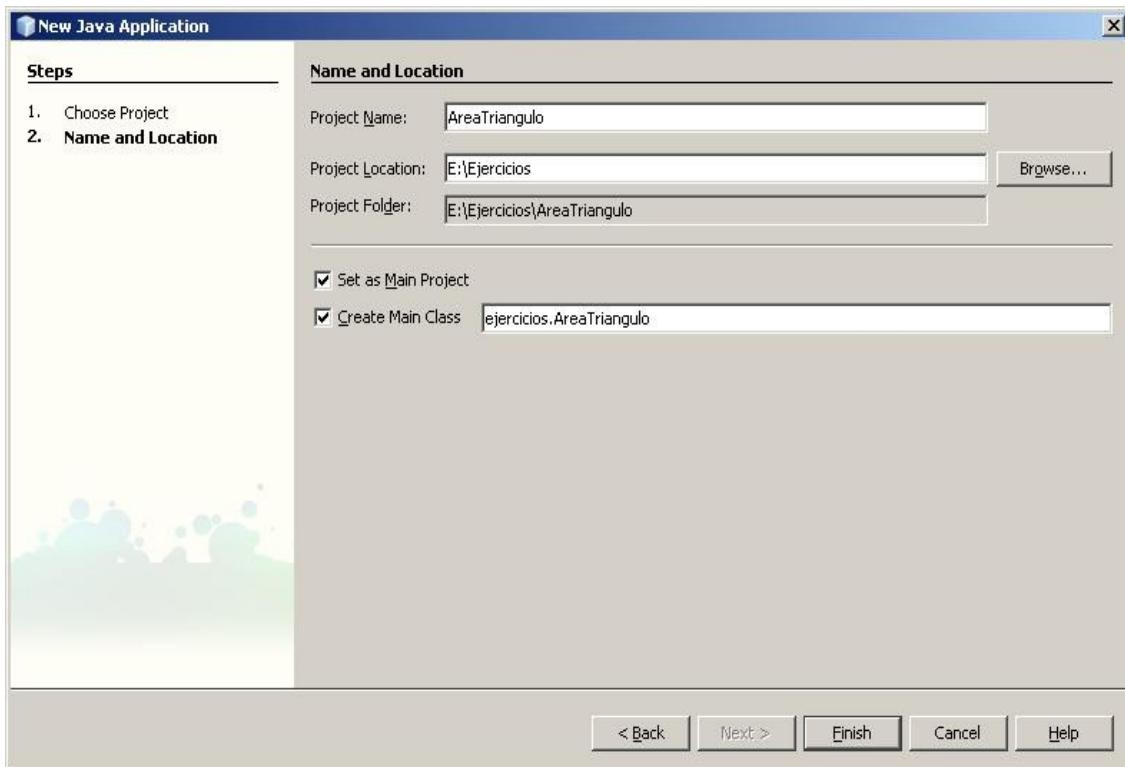


Seleccionamos para **Categories** Java y para Proyects **Java Application** y luego damos click en el botón de comando **Next**.



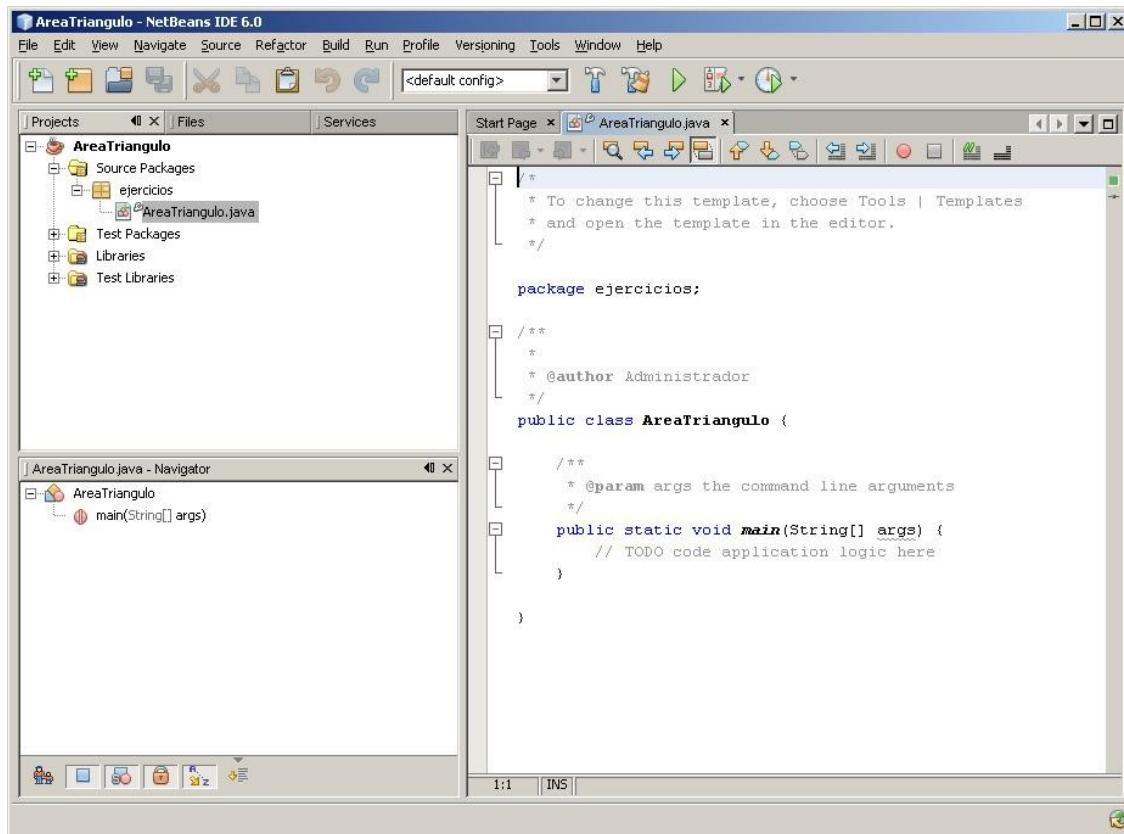
Usamos el botón de comando **Browse** para crear una carpeta denominada Ejercicios en la unidad E. En **Project Name** colocamos AreaTriangulo y en el cuadro de texto referido a **Create Main Class** colocamos *ejercicios.AreaTriangulo*, lo cual permitirá crear un paquete denominado *ejercicios* y como primera clase AreaTriangulo, es decir, se crea el archivo AreaTriangulo.java que pertenecerá al paquete *ejercicios*.





Sabemos que en Java una clase tiene comúnmente al método *Main* que es el método que se ejecuta cuando se aplica **Run** a la aplicación construida. En el entorno de NetBeans, una clase creada, crea un método que tiene el mismo nombre de la clase a la que se denomina *método constructor* y toda programación hecha en éste método se ejecutará primero antes que el método **main**. NetBeans generará un paquete (Package) denominado *ejercicios* y dentro de ella se mostrará la clase de acceso público *AreaTriangulo*. El entorno de desarrollo de NetBeans después de dar click en el botón **Finish** queda así:





Agregamos al código de la programación generada, por debajo del package ejercicios, lo siguiente:

```
import java.io.*;  
import javax.swing.*;
```

El paquete javax.swing.*; permitirá poder crear ingresos y salida de datos a través

de cajas de mensaje.

Agregamos al código en el método Main de la clase AreaTriangulo como se aprecia a continuación (lo escrito en azul).

```
package  
ejercicios;  
import java.io.*;  
import  
javax.swing.*;  
  
/**  
*  
* @author Administrador  
*/
```



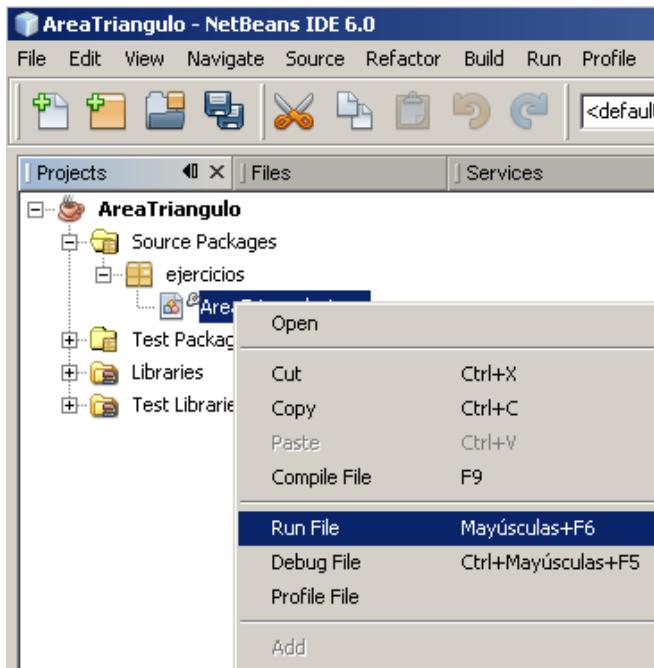


```
public class AreaTriangulo {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) throws IOException  
{  
        int base, altura;  
        double area;  
        base=Integer.parseInt(JOptionPane.showInputDialog(null,"Ingrese la base:"));  
        altura=Integer.parseInt(JOptionPane.showInputDialog(null,"Ingrese la altura"));  
        area=base*altura/2;  
        JOptionPane.showMessageDialog(null,"El area del triangulo es: "+area);  
    }  
  
}
```

En el método main observamos que para leer el dato para la variable **base** utilizamos el método *showInputDialog* de la clase *JOptionPane* que a su vez pertenece al paquete *swing*, que mostrará un caja de mensaje para la lectura de datos. De igual manera se trabajará para leer el dato de altura.

También se observa el método *showMessageDialog* de la clase *JOptionPane* que permite mostrar en una caja de mensaje el resultado del cálculo del área.

Seleccionamos *AreaTriangulo* en el entorno de desarrollo de NetBeans y luego damos click botón derecho del mouse. Se visualiza un menú flotante, se elige la opción **Run File** y se procederá a ejecutar el programa.



Cuando la aplicación es ejecutada se visualizará la siguiente ventana:



Nos pide el ingreso del valor de la base y luego de dar click en el botón de comando **Aceptar** se mostrará la siguiente ventana:



Posteriormente se mostrará la ventana del resultado del cálculo del área.





Objeto de control JLabel

Un objeto de control JLabel permite dibujar en el formulario una etiqueta, entendiéndose como etiqueta una expresión estética que se quiere colocar. También es usado para mostrar los resultados de un proceso.

Este componente se utiliza para desplegar textos o mensajes estéticos dentro de las formas, textos tales como encabezados, solicitud al usuario del programa para que proporcione algún dato o información (edad, dame sueldo, etc.), en cierta forma hace las funciones de printf, cout, writeln, print, display, etc., pero solo cuando se consideran en su parte de mensajes.

También es un objeto en java y por tanto tiene asociados sus propias propiedades y objetos-eventos, al mismo tiempo como se está usando dentro del objeto FRAME, muchas propiedades que se definan para el objeto FRAME, el objeto Label las va a heredar.

Es la propiedad TEXT, la que lleva el contenido del mensaje que se quiere desplegar, misma que se cargara de manera igual que con FRAME, es decir LABEL.setPROPIEDAD(parámetros, en muchos de los casos;

Propiedades más usadas:

Background(), Border(), Bounds(), Cursor(), Enabled(), Font(), Foreground(), Name(), Opaque(), Text(), ToolTipText(), Visible()

Text: Contiene el valor que se visualizará en el formulario.

Font: Permite establecer el tipo de letra de la expresión a mostrar en el formulario.

Border: Para establecer el tipo de borde de la etiqueta.

Enabled: Para habilitar o inhabilitar el uso del objeto de control.

Recordar que las propiedades se inicializan y se usan en código con .SETpropiedad.

Algunos de sus escuchadores con sus métodos mas importantes son:

WINDOW

: WindowActivated(), WindowClosed(), WindowClosing(), WindowDeactivated(),
WindowOpened().

MOUSE: MouseClicked(), MouseDragged(), MouseEntered(), MouseExited(),
MouseMoved(), MousePressed(), MouseReleased().

KEY: KeyPressed(), KeyReleased(), KeyTyped().





Objeto de control JTextField

Un objeto de control JTextField permite dibujar en el formulario un cuadro de texto, es decir, una caja que permite la introducción de un dato o valor. Este objeto es utilizado para el ingreso de datos.

En Java este componente es un importante componente visual, su función principal es manejar todos los procesos de entrada y salida (input/output) al programa.

Su principal propiedad es TEXT(), que es la propiedad que contiene el dato de tipo STRING que se manipulará en los programas o aplicaciones.

El dato dentro de esta propiedad es de tipo STRING y por tanto se tendrá que convertir a datos numéricos normales para poder efectuar procesos numéricos con ellos como se vio en el JAVA TEMA de CAST y CONVERSIONES.

Aparte de esta propiedad TEXT, tiene otras propiedades que se pueden inicializar con el comando SET como ya se ha visto en los JAVA TEMA s de FRAME y LABEL.

Propiedades más usadas:

Background(), Border(), Bounds(), Columns(), Cursor(), Editable(), Enabled(), Font(), Foreground(), Name(), Opaque(), Text(), ToolTipText(), Visible()

Text: Contiene el valor o dato introducido en el cuadro de texto.

Font: Permite establecer el tipo de letra del texto en la caja.

Border: Para establecer el tipo de borde del cuadro de texto.

Enabled: Para habilitar o inhabilitar el uso del objeto de control.

Métodos más usados:

Recordar que las propiedades se inicializan y se usan en código con .SETpropiedad.

WINDOW: WindowActivated(), WindowClosed(), WindowClosing(), WindowDeactivated(), WindowOpened().

MOUSE: MouseClicked(), MouseDragged(), MouseEntered(), MouseExited(), MouseMoved(), MousePressed(), MouseReleased().

KEY: KeyPressed(), KeyReleased(), KeyTyped().

getText(): Permite obtener el texto introducido en el cuadro de texto.

setText(): Permite colocar un texto en el objeto JTextField.

requestFocus(): permite asignar el cursor al objeto de control.





Objeto de control JButton

Un objeto de control JButton permite dibujar en el formulario un objeto que contiene un proceso a ejecutar. Se utiliza comúnmente para llevar a cabo procesos específicos según la naturaleza de la aplicación.

En java es el control principal de la forma, contiene el código principal del programa y su activación por el usuario provoca que se realicen los principales procesos del problema planteado (aquí es donde se capturan datos, se realizan operaciones, etc.).

De este control se maneja su propiedad TEXT para etiquetarlo con la palabra "OK" o "ACEPTAR" o "EXE" y su evento Click para activarlo, es en dicho evento donde se construye el código del programa.

Propiedades más usadas:

Background(), Border(), Bounds(), Columns(), Cursor(), Editable(), Enabled(), Font(), Foreground(), Name(), Opaque(), Text(), ToolTipText(), Visible()

Text: Contiene el valor o dato introducido en el cuadro de texto.

Font: Permite establecer el tipo de letra del texto en la caja.

Enabled: Para habilitar o inhabilitar el uso del objeto de control.

Evento más usado:

ActionPerformed: Este evento se lleva a cabo cuando el usuario da click sobre el objeto de control JButton.

WINDOW: WindowActivated(), WindowClosed(), WindowClosing(), WindowDeactivated(), WindowOpened().

MOUSE: MouseClicked(), MouseDragged(), MouseEntered(), MouseExited(), MouseMoved(), MousePressed(), MouseReleased().

KEY: KeyPressed(), KeyReleased(), KeyTyped().

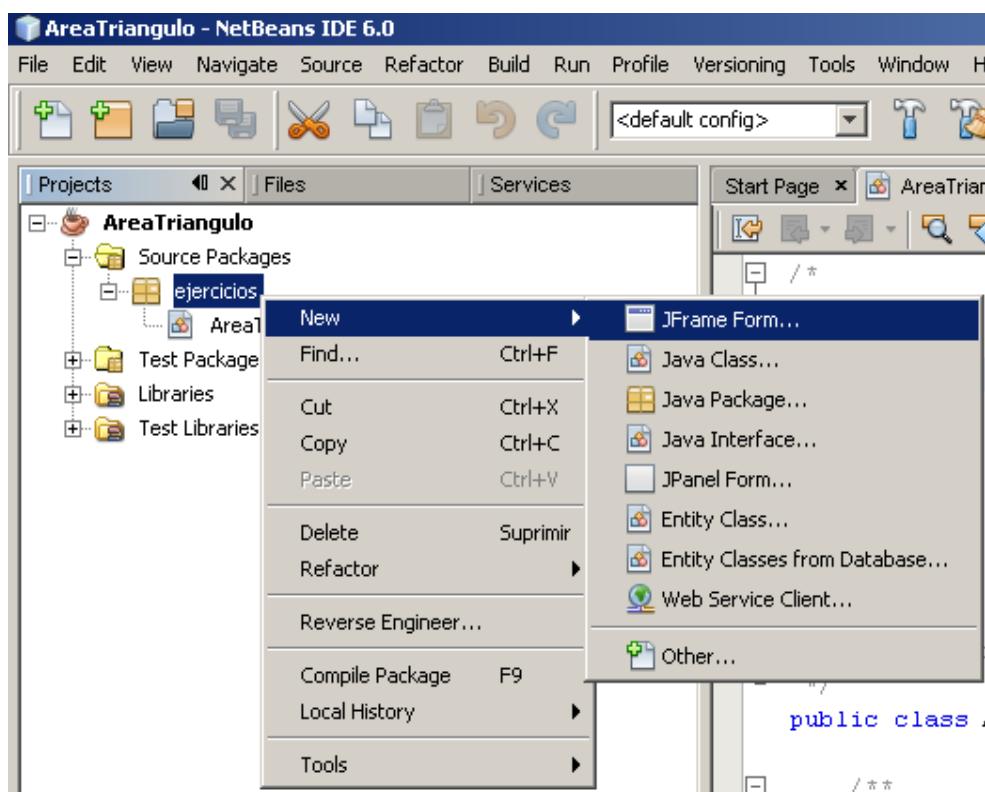


Una aplicación usando los objetos de control

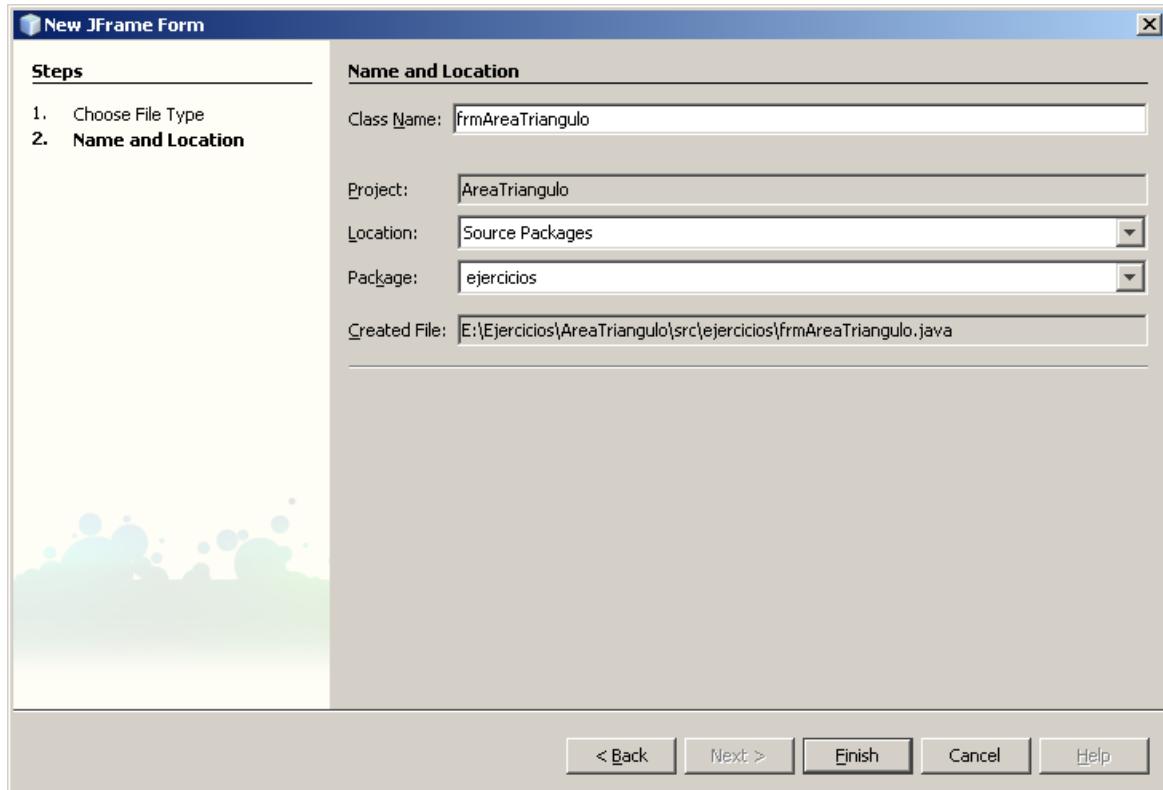
Ahora procedamos a desarrollar la misma aplicación usando como interfase un formulario y los objetos de control antes mencionado.

Solución:

Seleccionamos el paquete ejercicios y damos click botón derecho del mouse y elegimos la opción **New** y posteriormente **JframeForm**.



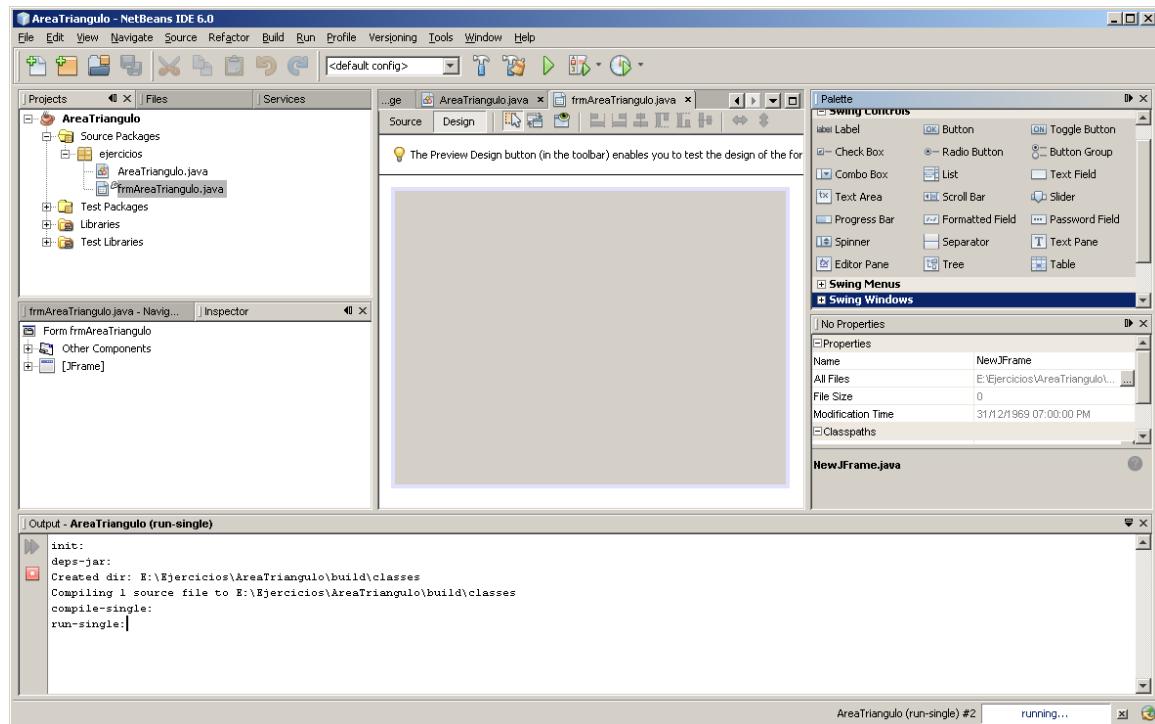
Se muestra la ventana **New JFrame Form** y colocamos en Class Name el nombre del formulario: **frmAreaTriangulo**.



En esta ventana se observa que la clase denominada **frmAreaTriangulo** generará un archivo de extensión .java denominado **frmAreaTriangulo** que se almacenará dentro de la carpeta **ejercicios** y pertenecerá al paquete **ejercicios**.

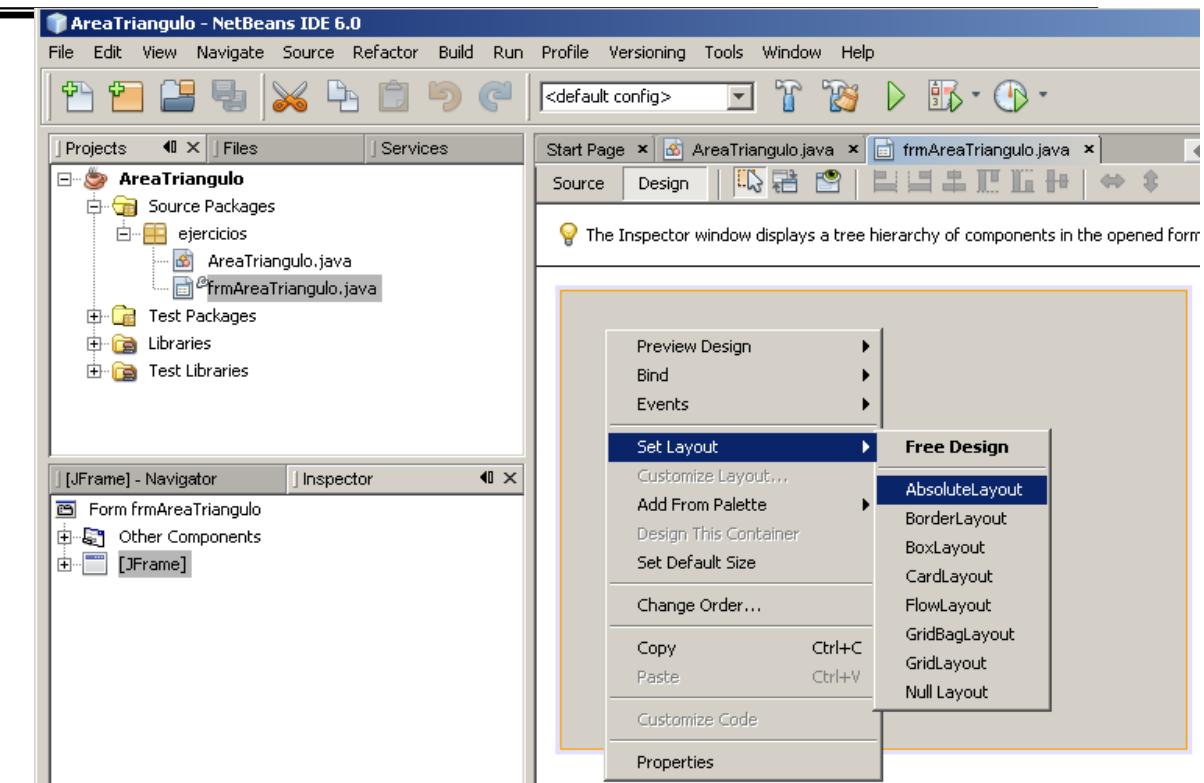
Al momento de dar click en el botón de comando **Finish** se visualizará el entorno de desarrollo NetBeans y al lado derecho se muestra la paleta de los objetos de control (Swing Controls).





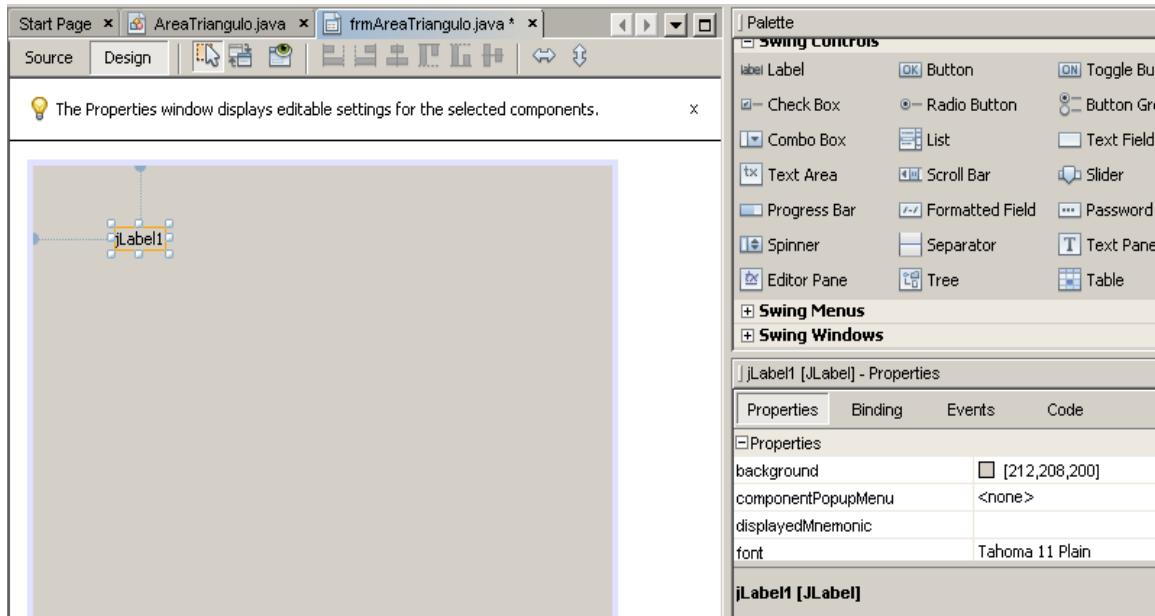
Sobre el diseño del formulario damos click botón derecho y seleccionamos **Set Layout** y posteriormente **AbsoluteLayout**.



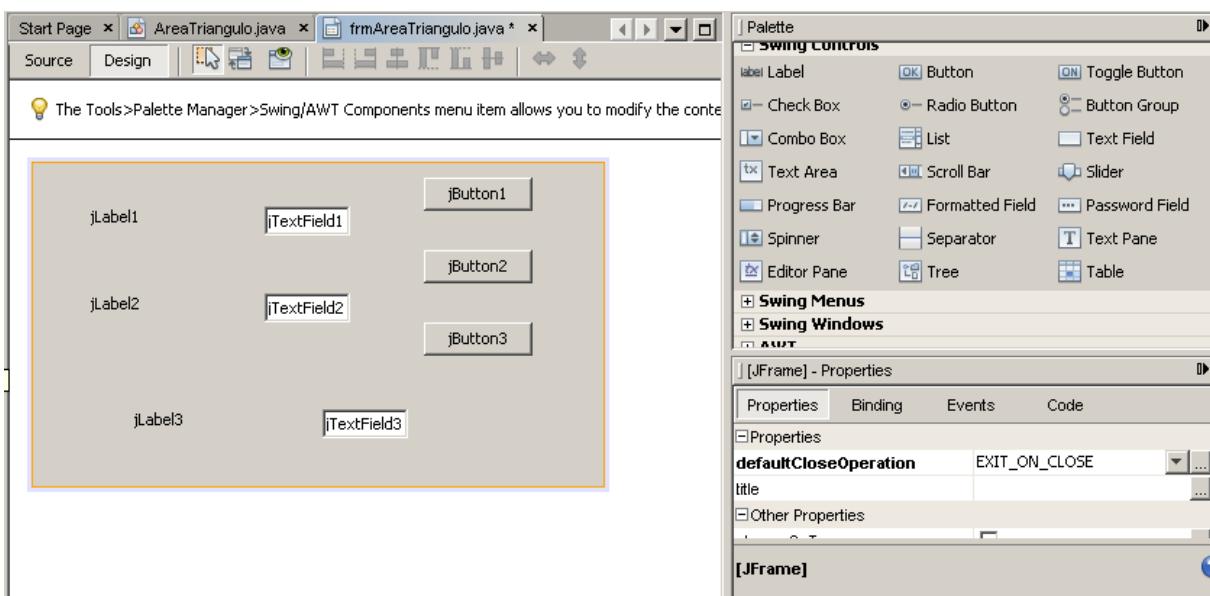


Es necesario usar **AbsoluteLayout** para que permita dibujar los objetos de control en el lugar donde uno quiere en el formulario.

Ahora procedamos a colocar un objeto **JLabel** seleccionando de la paleta Swing **Controls Label** y lo arrastramos hacia el diseño del formulario.



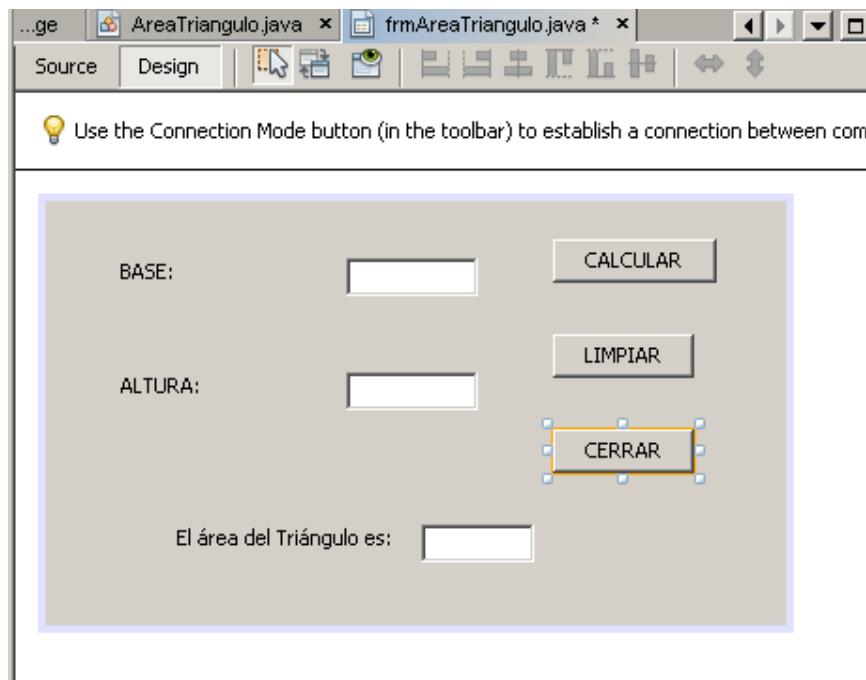
Continuamos el diseño del formulario, quedando éste de la siguiente manera:



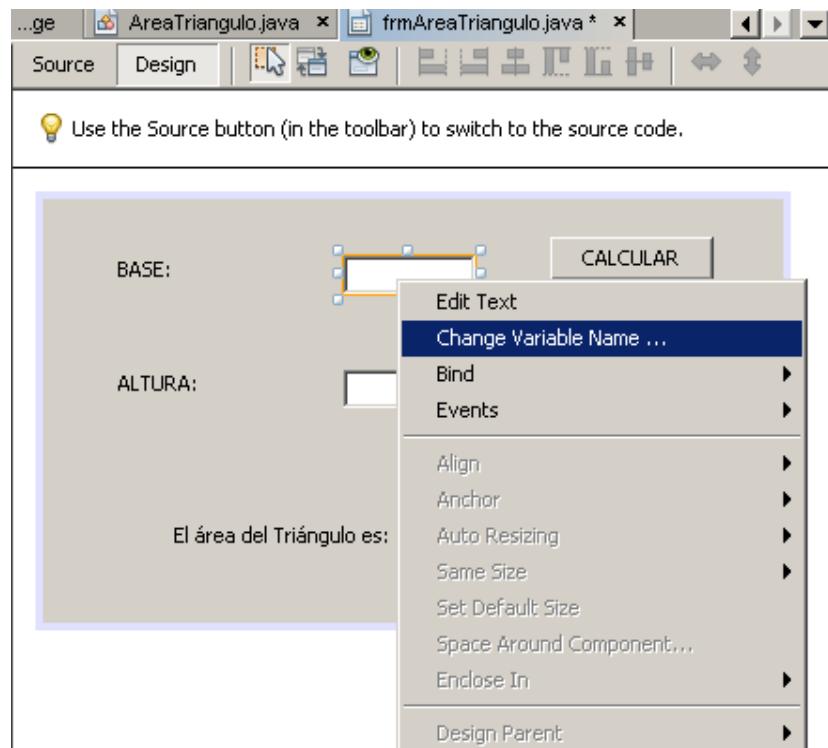
Ahora procedamos a usar a cambiar los valores de las propiedades de los objetos de control en la ventana de propiedades:

Objeto de Control	Propiedad	Valor de la Propiedad
JLabel1	Text	BASE:
JLabel2	Text	ALTURA:
JLabel3	Text	El área del Triángulo es:
JTextField1	Text	(Vacío o limpiar)
JTextField2	Text	(Vacío o limpiar)
JTextField3	Text	(Vacío o limpiar)
JButton1	Text	CALCULAR
JButton2	Text	LIMPIAR
JButton3	Text	CERRAR

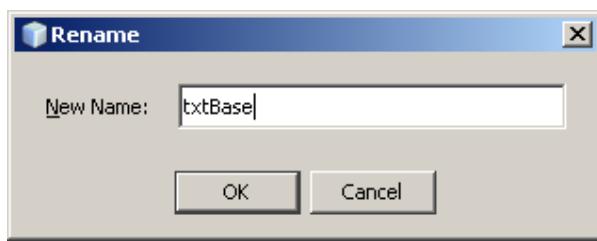
Luego de aplicar los cambios en los valores de propiedades el diseño del formulario debe quedar de la siguiente manera:



Es necesario darle un nombre propio a los objetos de control y sobretodo a aquellos que intervienen en la lógica de la programación. Seleccionamos el objeto de control JTextField1 y damos click con el botón derecho del mouse y se visualizará un menú flotante y seleccionamos **Change Variable Name**.



Colocamos como nombre al objeto JTextField1: **txtBase**





Continuamos con los objetos de control siguientes:

Objeto de Control	Nombre
JTextField2	txtAltura
JTextField3	txtArea
JButton1	btnCalcular
JButton2	btnLimpiar
JButton3	btnCerrar

Ahora procedamos a programar en los botones de comando.

En el botón CALCULAR (btnCalcular) al darle doble click y escribir el siguiente código: (lo escrito en azul)

```
private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt)
{
    int base, altura;
    double area;
    base=Integer.parseInt(txtBase.getText());
    altura=Integer.parseInt(txtAltura.getText());
    area=base*altura/2;
    txtArea.setText(String.valueOf(area));
}
```

Una vez declaradas las variables de memoria, a la variable **base** se le asigna el valor introducido en el cuadro de texto **txtBase**. El método **getText()** permite obtener el dato introducido y con el método **parseInt** de la clase Integer es convertido a numérico entero. Se hace lo mismo para la variable altura. Para mostrar el cálculo de área se utiliza el método **setText** del cuadro de texto txtArea que permite visualizar el contenido de la variable **area**. A la variable se le aplica el método **valueOf** de la clase String para convertir el dato área en cadena de caracteres.





En el botón LIMPIAR (btnLimpiar) luego de darle doble click escribimos el siguiente código: (lo escrito en azul)

```
private void btnLimpiarActionPerformed(java.awt.event.ActionEvent evt)
{
    txtBase.setText("");
    txtAltura.setText("");
    txtArea.setText("");
    txtBase.requestFocus();
}
```

Se limpian los cuadros de textos a través del método `setText()` y con el método `requestFocus()` se pasa el cursor al objeto de control `txtArea`.

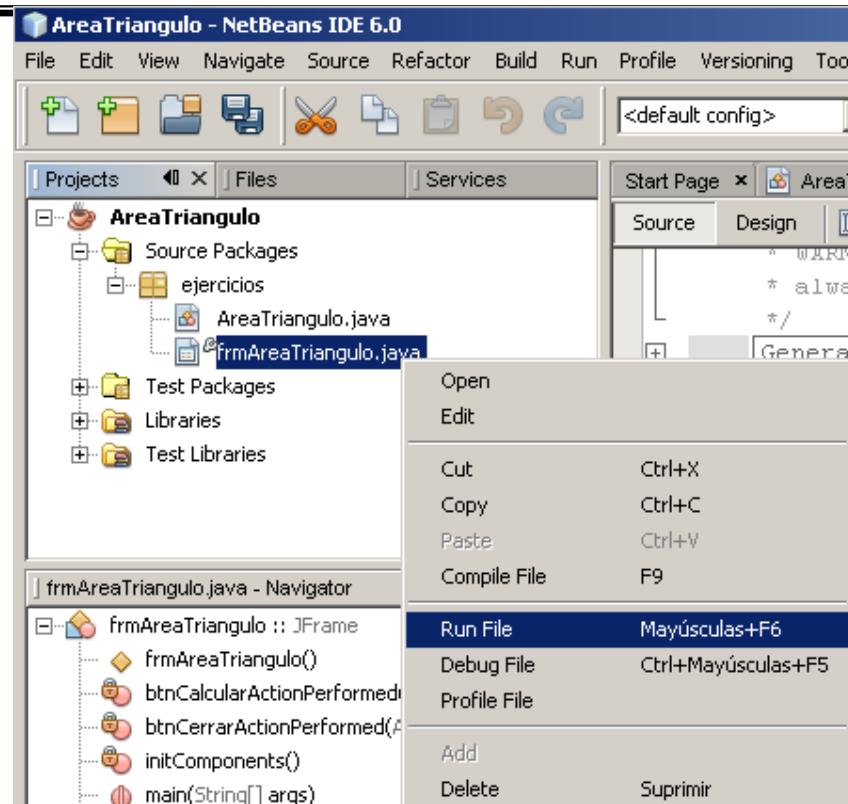
En el botón CERRAR (btnCerrar) luego de darle doble click, escribimos el siguiente código: (lo escrito en azul)

```
private void btnCerrarActionPerformed(java.awt.event.ActionEvent evt)
{
    dispose();
}
```

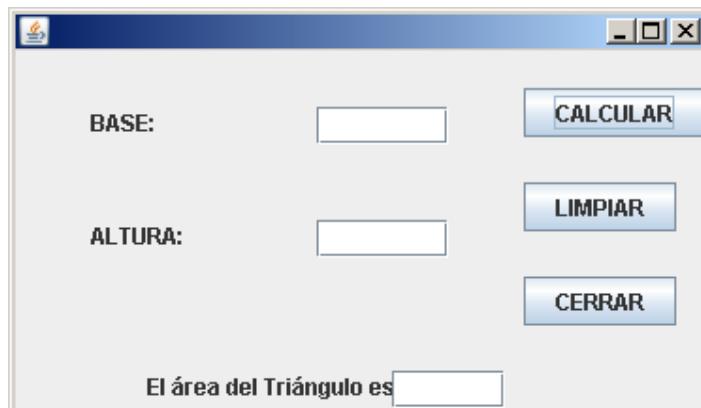
El método `dispose()` permite descargar el formulario y terminar la ejecución de la aplicación.

Procedemos a ejecutar la aplicación seleccionado `frmAreaTriangulo` y al dar clic botón derecho elegimos **Run File**.





Se muestra el formulario diseñado en etapa de ejecución.



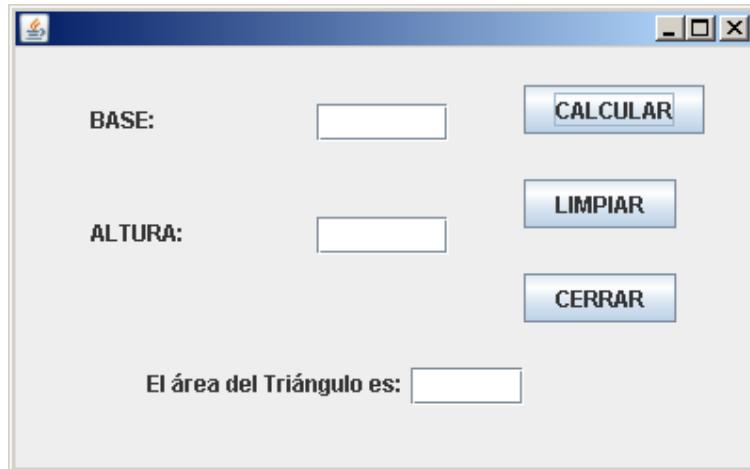
Podemos observar que el formulario sale con los objetos de control cercano a los bordes del formulario y el mismo formulario se visualiza pegado en la parte superior izquierda de la pantalla del computador. Vamos a proceder a corregir estos defectos agregando dos líneas de código en el método constructor de la clase frmAreaTriangulo. (escribe lo que está en azul)



```
public class frmAreaTriangulo extends javax.swing.JFrame
{
    /** Creates new form frmAreaTriangulo */
    public frmAreaTriangulo()
    {
        initComponents();
        setSize(400,250);
        setLocation(250,250);
    }
}
```

El método `setSize()` permite establecer el tamaño del formulario y el método `setLocation()` permite ubicar el formulario dentro de la pantalla. Los métodos antes mencionados pertenecen al formulario `frmAreaTriangulo`. Otros métodos del formulario se verán más adelante.

Procedemos nuevamente a ejecutar el formulario `frmAreaTriangulo` y se mostrará de la siguiente manera:



En el siguiente tema conoceremos más objetos de control y seguiremos trabajando con lo aprendido en esta sesión o tema.

