

Creación de Menús.

Los menús son muy utilizados en el desarrollo de interfaces graficas, Java provee varios controles que permiten hacer uso de Barras de menús estos controles son:

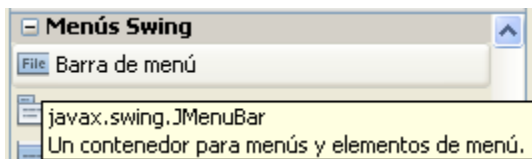
JMenuBar: corresponde a la clase utilizada para crear una barra de menú principal, gráficamente es una barra horizontal alargada que contiene los diferentes menús.

JMenu: Es una clase que permite agregar otros menús a la barra del JMenuBar. Esto permitirá agregar más opciones a un menú

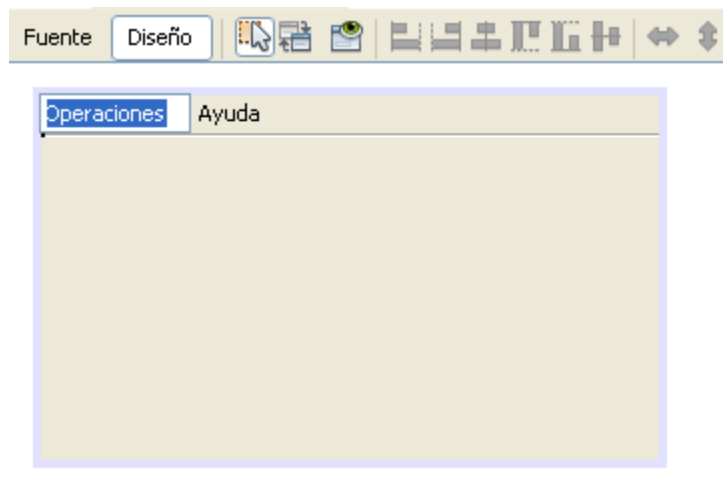
JMenuItem: Es la clase que permite agregar una opción del JMenu.

Ejercicio de Aplicación

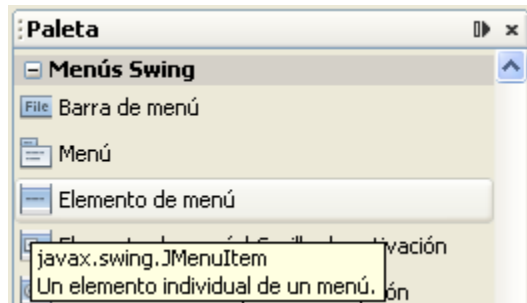
1. Para agregar un menú al formulario seleccione en la Paleta de Componentes la sección Menús Swing y haga clic en el control Barra de menú (JMenuBar) y luego en el formulario.



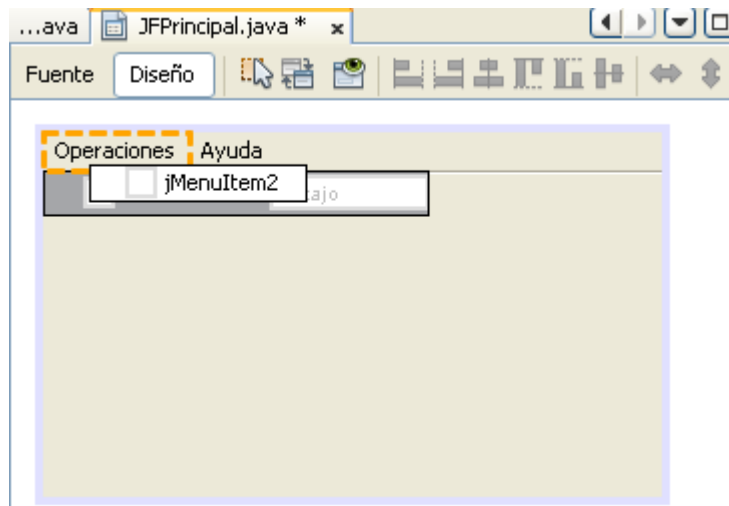
2. En el menú que aparece podrá modificar los nombres de los menús haciendo doble clic en cada una y digitando el nuevo contenido.



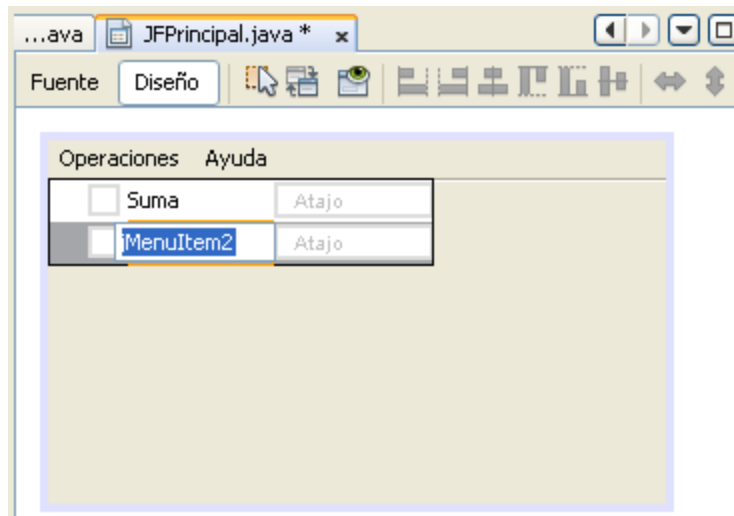
3. Para agregar opciones a cada menú seleccione en la Paleta de controles la opción Elemento de menú (JMenuItem).



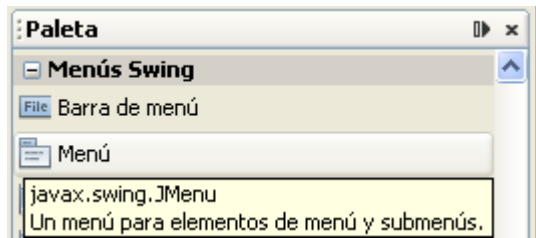
4. Luego arrastre el cursor hasta que aparezca un cuadro punteado sobre la opción donde desea agregar el menú.



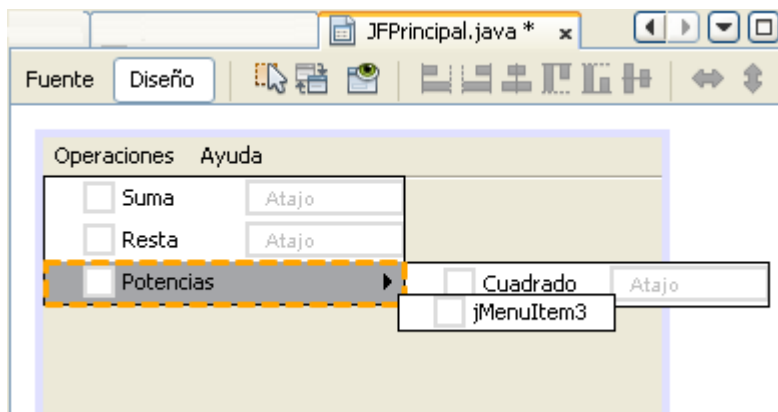
5. De la misma forma en que modifiko los nombres de los menús puede cambiar las opciones agregadas.



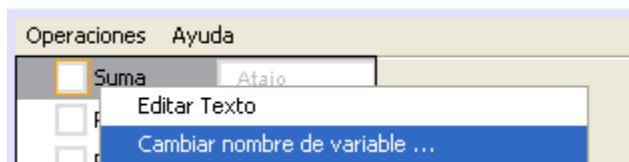
- Si desea agregar opciones que tienen subopciones, adicione controles de tipo Menú (JMenu) y dentro de estos controles de Elemento de menú (JMenuItem)



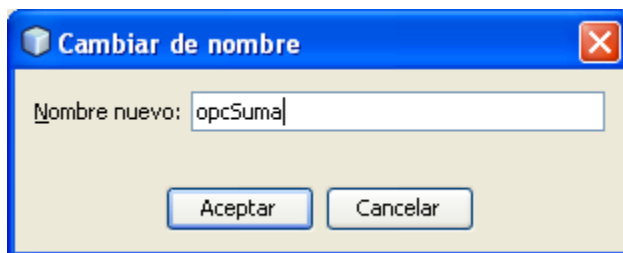
Resultado de agregar opciones con subopciones.



- Para cambiar los nombres de los controles y facilitar la codificación de los mismos, solo se debe hacer clic derecho sobre la opción y seleccionar Cambiar nombre de variable...



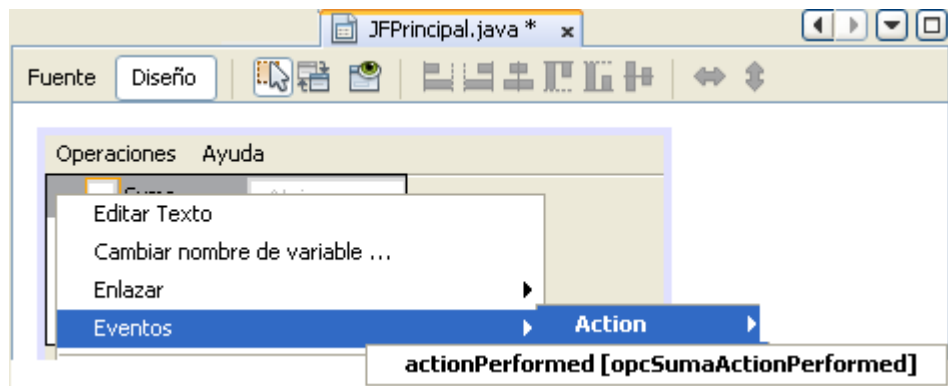
Luego digite el nombre del control.



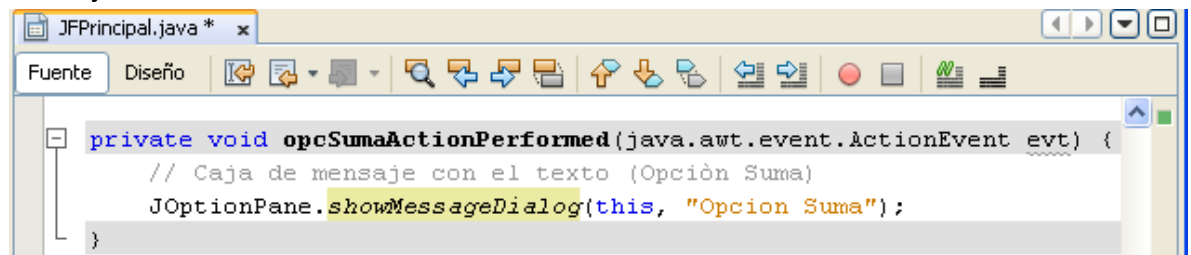
8. Para codificar cada opción, usted podrá codificar directamente en el formulario o crear un controlador, los dos procesos se explican a continuación:

- **Codificar Directamente En El Formulario**

Haga clic derecho sobre la opción que desea programar y en el menú que aparece seleccione la opción *Eventos* luego *Action* y aparecerá la opción `actionPerformed[nombre de opción de menú]` (también puede hacer doble clic sobre la opción).



Automáticamente podrá observar la ventana de codificación del formulario y el método correspondiente (Fuente), donde podrá agregar las instrucciones que desee ejecutar cuando el usuario seleccione dicha opción. Por ejemplo, una caja de mensaje.



- **Crear Un Controlador**

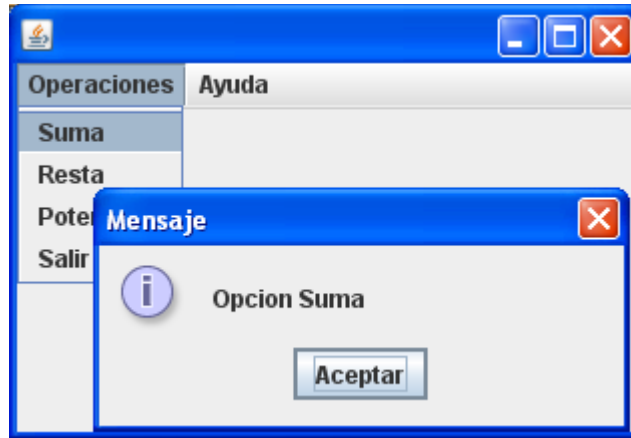
Cree los métodos get y set de cada opción del menú y luego cree una clase llamada Controlador y agregue el siguiente código:

```
/**
 * @author Sonia Pinzón
 */
public class Controlador implements ActionListener{
    frmMenu frmM;
    public Controlador() {
        frmM= new frmMenu();
        frmM.getOpcSuma().addActionListener(this);
        frmM.getOpcResta().addActionListener(this);
        frmM.getOpcCuadrado().addActionListener(this);
    }
    public void iniciar(){
        frmM.setVisible(true);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getSource()== frmM.getOpcSuma()){
            JOptionPane.showMessageDialog(frmM,"Opción Suma");
        }
        if(e.getSource()==frmM.getOpcResta()){
            JOptionPane.showMessageDialog(frmM,"Opción Resta");
        }
        if(e.getSource()==frmM.getOpcCuadrado()){
            JOptionPane.showMessageDialog(frmM,"Opción Resta");
        }
    }
}
```

Posteriormente cree una clase Main, y agregue el siguiente código, con el cual podrá ejecutar el controlador.

```
package vista;
import control.Controlador;
public class ProgPrincipal {
    public static void main(String[] args) {
        Controlador objC= new Controlador();
        objC.iniciar();
    }
}
```

9. De esta manera podrá ejecutar el formulario y luego seleccionar las opciones requeridas para obtener la respuesta correspondiente. En este caso si el usuario selecciona la opción Suma del menú Operaciones se visualizará la siguiente Caja de mensaje.



10. Si se desea validar la opción salir, se deberá agregar el siguiente código:

Para crear una caja de mensaje que confirme la salida del programa se debe hacer uso del método `showConfirmDialog()` de la clase `JOptionPane`. Este método requiere tres parámetros:

`JOptionPane.showConfirmDialog(contenedor,texto,titulo, tipo_botones,tipo_caja);`

Contenedor: hace referencia al objeto donde se visualizará la caja de mensaje, en este caso se utiliza el apuntador `this` para indicar que se visualizará sobre el formulario actual.

Texto: corresponde al texto que se visualizará dentro de la caja de mensaje.

Título: modifica el título de la caja de mensaje.

tipo_botones: es el parámetro que determina el tipo de botones que aparecerá en la caja de mensaje, por ejemplo: botones Aceptar/Cancelar o botones Si/No.

tipo_caja: define el tipo de mensaje se presentará en la caja, por ejemplo: `QUESTION_MESSAGE` (mensaje de pregunta).

```
private void opcSalirActionPerformed(java.awt.event.ActionEvent evt) {  
    int resp= JOptionPane.showConfirmDialog(this, "Esta seguro de salir?" ,  
        "Confirmar",JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);  
    if (resp==JOptionPane.YES_OPTION )  
    { this.dispose();  
    }  
}
```

Al seleccionar la opción salir se observará la siguiente caja de dialogo.

