

**TECNOLOGICO NACIONAL DE MEXICO
CAMPUS "OAXACA"**

INSTITUTO TECNOLÓGICO DE OAXACA

INGENIERIA EN SISTEMAS COMPUTACIONALES.

NOMBRE DE LA MATERIA: TOPICOS AVANZADOS DE PROGRAMACION.

UNIDAD 1: "INTERFAZ GRAFICA DEL USUARIO".

ACTIVIDAD 2: GUIA 2 DE INTERFAZ DE USUARIO.

GRUPO: ISU

HORA: 09:00-10:00 AM

DOCENTE: M.C. LUIS ALBERTO ALONSO HERNÁNDEZ

ALUMNO: GONZALEZ PASCUAL MELVIN PAUL

FECHA DE ENTREGA: 04 DE OCTUBRE DEL 2020

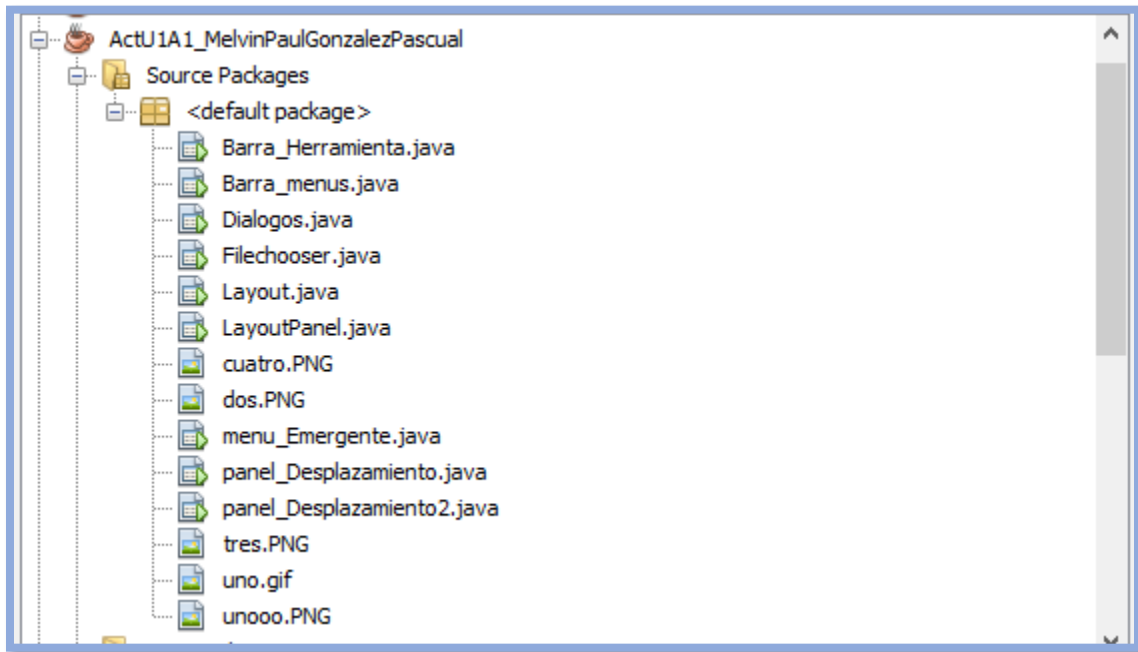
Contenido

Introducción:	3
Menús.....	4
Barras de herramientas.....	6
Menús emergentes.	8
Filechooser.	10
ScrollPane.....	12
Variables globales.....	14
Centralizar código.	17
Layaouts.	21
Layouts y paneles.	23
Diálogos.....	25

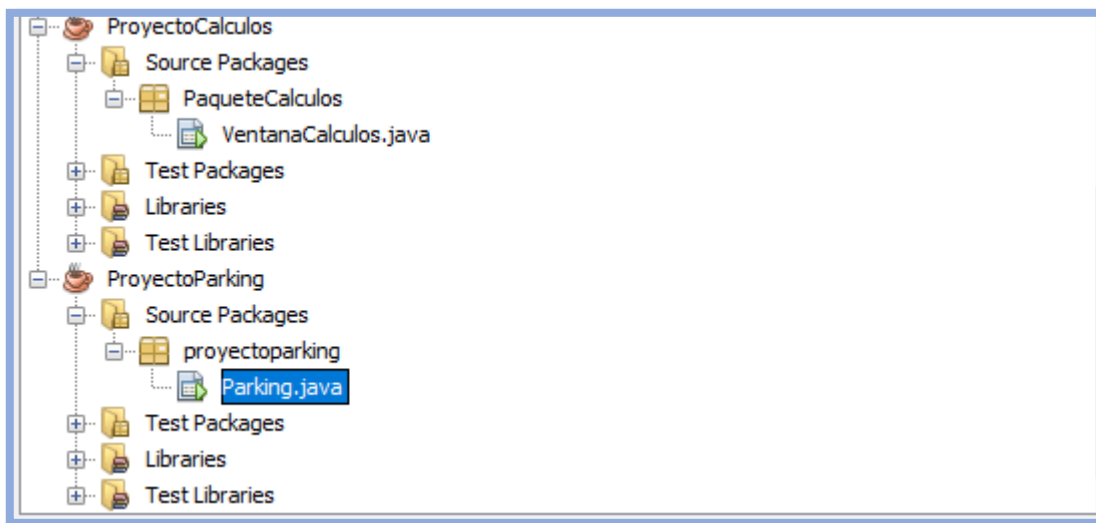
Introducción:

En esta segunda actividad se desarrollarán del componente a una mayor funcionalidad, las cuales se usan los componentes de JFrame, al igual que los diseños propios que se puede llegar a tener a continuación se explicara como están distribuidos los proyectos debido a que en algunas prácticas te pedían realizarlo.

Nuestro primer proyecto con el nombre de ActU1A1_MelvinPaulGonzalezPascual, tendrá almacenado las siguientes clases.



Los siguientes dos proyectos son propios ya que en las actividades pedían realizar de esa manera.

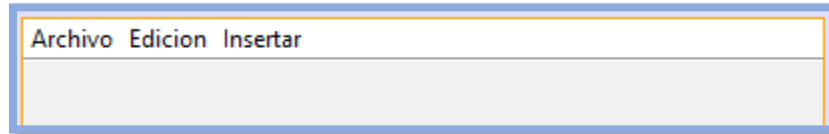


Se explicará brevemente el funcionamiento de cada práctica.

Menús.

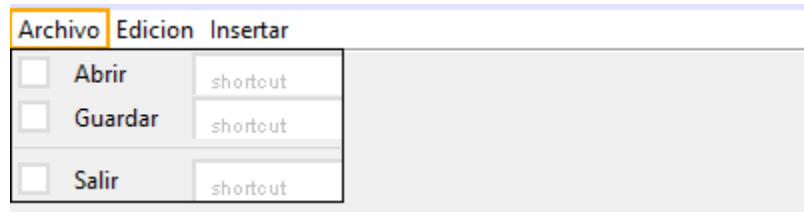
Para esta actividad conoceremos los componentes de menús, los cuales son JMenuBar, JMenu, JMenuItem y Separator los cuales son esenciales para la creación de los menús en nuestros programas.

Lo primero a crear es un menú de barras como se muestra a continuación.



Lo siguientes será crear un menú en las opciones de Archivo y Edición.

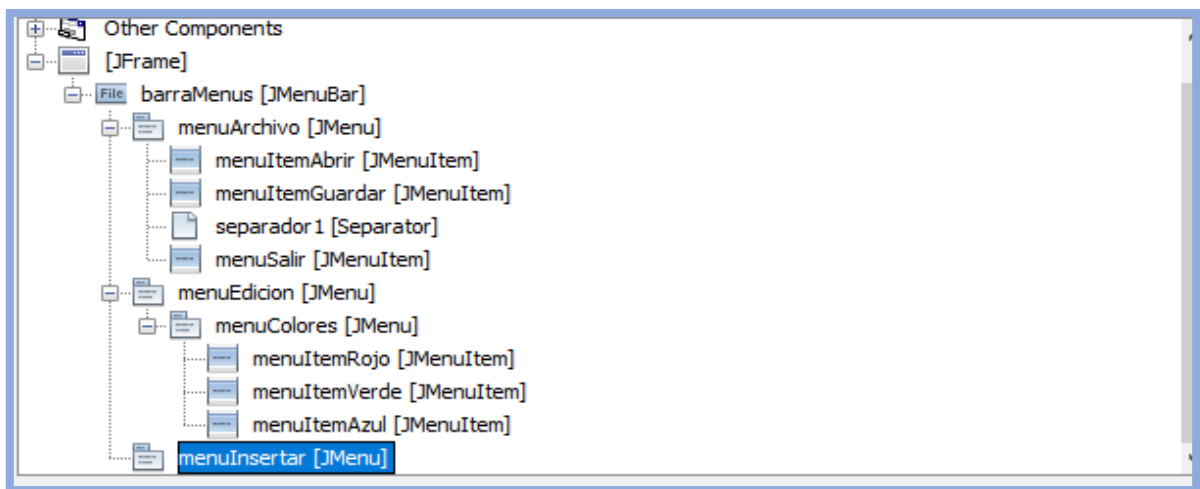
En la sección de archivo de nuestro Menú, tendrá 3 JMenuItem, lo cual quedará de la siguiente forma.



En la sección de Edición se creará un JMenuItem el cual contendrá, 3 JMenuItem, en el cual trabajaremos en esta actividad.



Nuestra lista de componentes que se llegaron ocupar queda la siguiente manera.



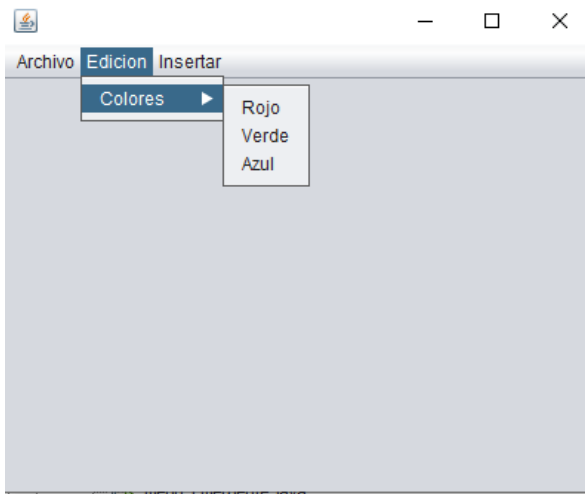
En la parte de código hay que tener en cuenta que se usara la siguiente librería.

```
import java.awt.Color;
```

El siguiente código, se utilizará en nuestros menús, Rojo, Verde y Azul, que quedaran de la siguiente manera.

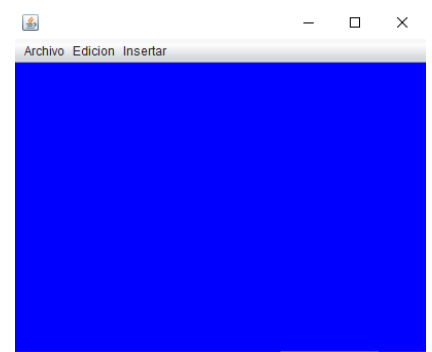
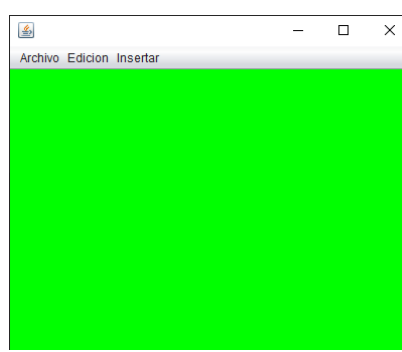
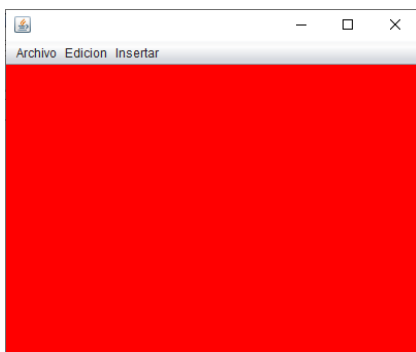
```
private void menuItemAzulActionPerformed(java.awt.event.ActionEvent evt) {  
    this.getContentPane().setBackground(Color.BLUE);  
    // TODO add your handling code here:  
}  
  
private void menuItemRojoActionPerformed(java.awt.event.ActionEvent evt) {  
    this.getContentPane().setBackground(Color.RED);  
    // TODO add your handling code here:  
}  
  
private void menuItemVerdeActionPerformed(java.awt.event.ActionEvent evt) {  
    this.getContentPane().setBackground(Color.GREEN);  
    // TODO add your handling code here:  
}
```

Lo que realiza es cambiar de color del fondo a seleccionar en el menú creado como se muestra a continuación.



Así se ve cuando no hemos seleccionado ningún color dentro de nuestra opción.

En los diferentes colores seleccionados como se ve a continuación:



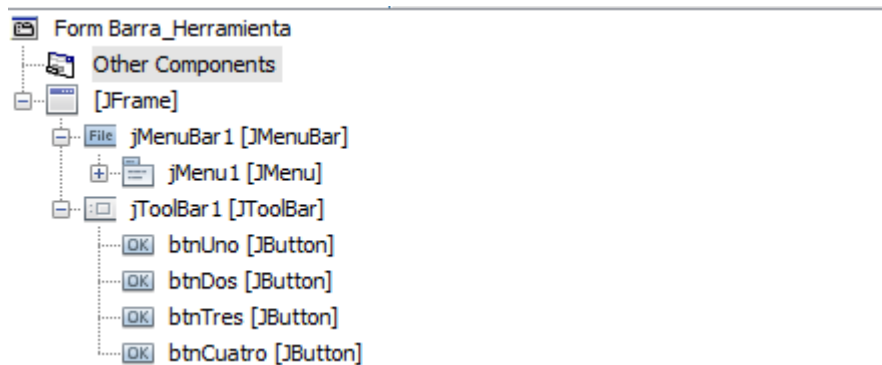
Barras de herramientas.

En esta actividad se usarán los componentes del menú, al igual que el del JToolBar, por lo cual se implementará el siguiente diseño.



En la imagen se muestra que contiene número, lo cual fue agregado con la propiedad de icon que traen los JButtons, el cual quedo de esta manera.

Los componentes que usaron fueron los siguientes los cuales se renombraron para un mejor funcionamiento.



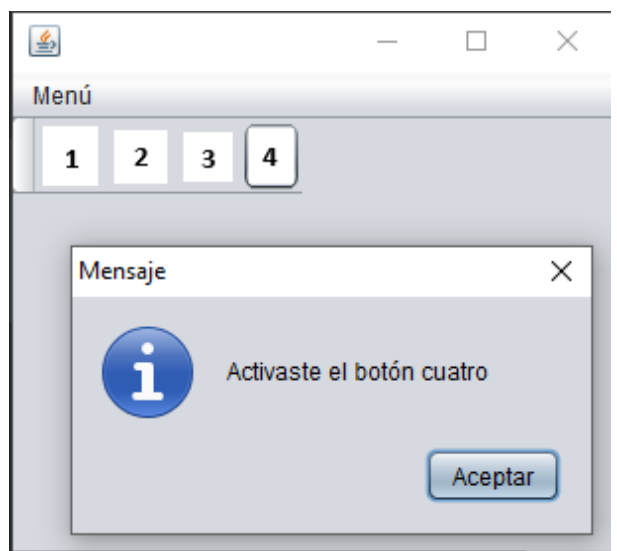
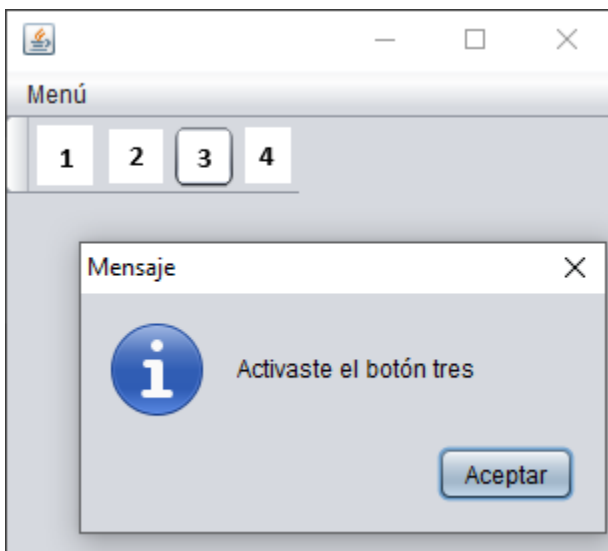
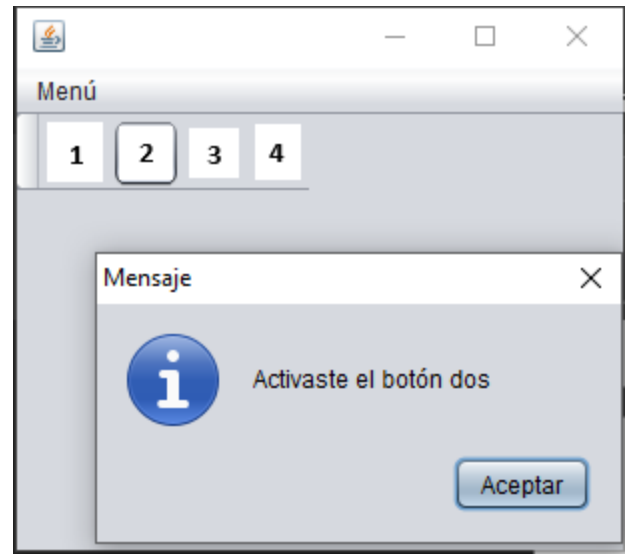
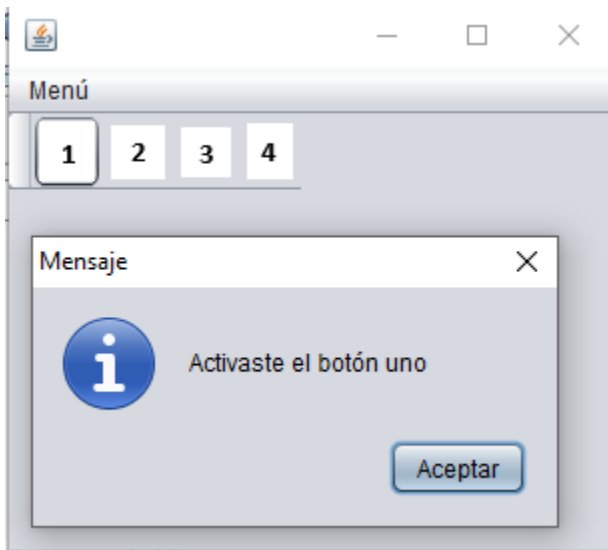
Todo el funcionamiento quedara en el control de los botones, los cuales deben contener el siguiente código.

```
private void btnUnoActionPerformed(java.awt.event.ActionEvent evt) {  
    JOptionPane.showMessageDialog(null,"Activaste el botón uno");  
    // TODO add your handling code here:  
}  
  
private void btnDosActionPerformed(java.awt.event.ActionEvent evt) {  
    JOptionPane.showMessageDialog(null,"Activaste el botón dos");  
    // TODO add your handling code here:  
}  
  
private void btnTresActionPerformed(java.awt.event.ActionEvent evt) {  
    JOptionPane.showMessageDialog(null,"Activaste el botón tres");  
    // TODO add your handling code here:  
}  
  
private void btnCuatroActionPerformed(java.awt.event.ActionEvent evt) {  
    JOptionPane.showMessageDialog(null,"Activaste el botón cuatro");  
    // TODO add your handling code here:  
}
```

Se debe recordar que necesario la siguiente librería para no tener algún error.

```
import javax.swing.JOptionPane;
```

Lo que realiza es mandarnos un panel con el texto de que botón fue presionado como se mostrara a continuación.



Como se puede observar cada que se presiona un botón nos lanzara una ventana con el resultado del botón presionado.

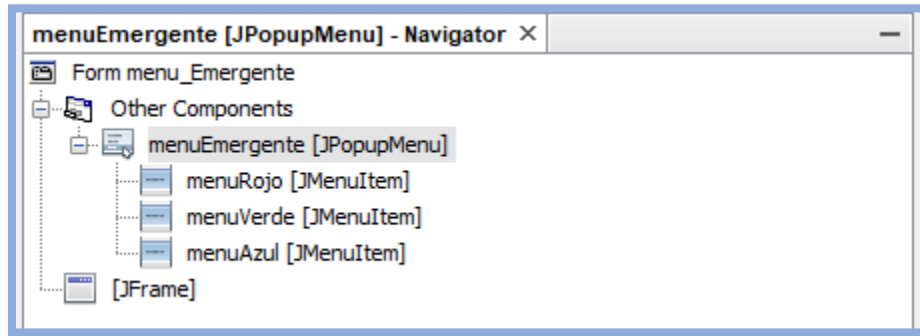
Por lo cual concluimos la actividad.

Menús emergentes.

En esta actividad se usará el componente JPopupMenu, el cual a diferencia de los demás, no aparece como tal en nuestro JFrame, ya aparece en la parte de otros componentes, al cual se le puede agregar distintos componentes como se muestra a continuación.

La ventaja es que cualquier

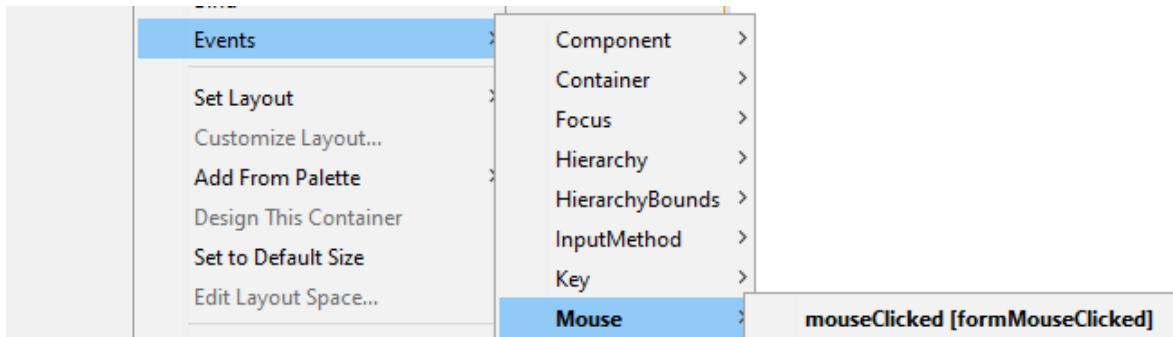
Los cuales le cambiara el nombre la variable para un mejor funcionamiento.



Es importante recordar que se tiene que importar la siguiente librería.

```
import java.awt.Color;
```

Antes de empezar se debe activar el evento de MouseClicked, lo cual nos ayudará para que en cualquier parte nuestro Frame nos salga nuestro menú.



El código para el evento es el siguiente.

```
private void formMouseClicked(java.awt.event.MouseEvent evt) {  
    menuEmergente.show(this, evt.getX(), evt.getY());  
    // TODO add your handling code here:  
}
```


Lo siguiente será codificar las acciones de nuestras opciones que en nuestro caso son Rojo, Verde y Azul, de la siguiente manera.

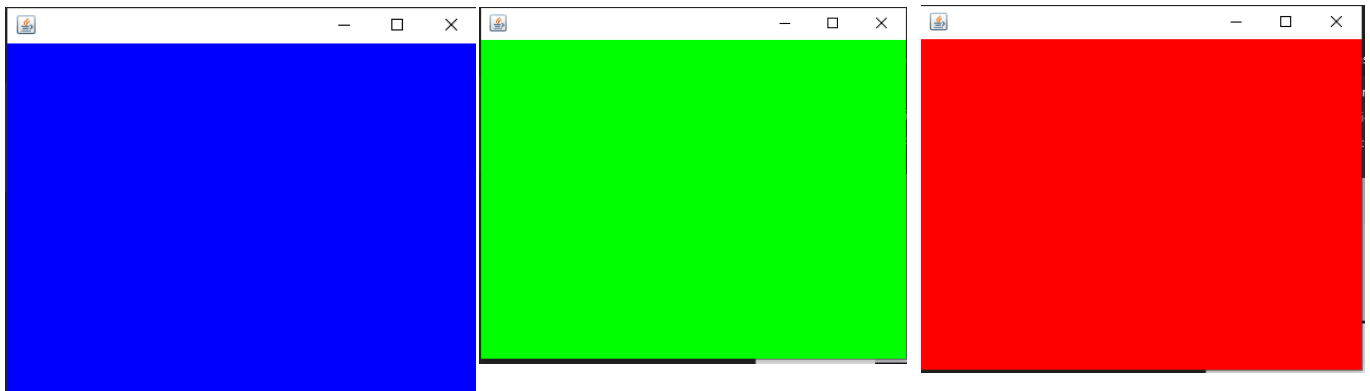
Lo que hace es poner el fondo el color seleccionado.

```
private void menuRojoActionPerformed(java.awt.event.ActionEvent evt) {  
    this.getContentPane().setBackground(Color.RED);  
    // TODO add your handling code here:  
}  
  
private void menuVerdeActionPerformed(java.awt.event.ActionEvent evt) {  
    this.getContentPane().setBackground(Color.GREEN);  
    // TODO add your handling code here:  
}  
  
private void menuAzulActionPerformed(java.awt.event.ActionEvent evt) {  
    this.getContentPane().setBackground(Color.BLUE);  
    // TODO add your handling code here:  
}
```

Como se muestra a continuación.



Como se puede observar al dar clic en el panel nos saldrá un menú, el cual hará la función de cambiar nuestro fondo.



Como se pudo observar el fondo cambia a los tres colores que tenemos como opción.

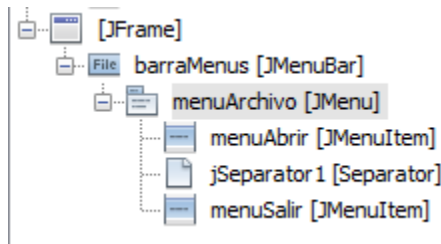
Filechooser.

Para esta actividad se usará el componente de JFileChooser el cual nos ayuda en varios programas para seleccionar algún documento de nuestro equipo por lo cual se implementa con menús.

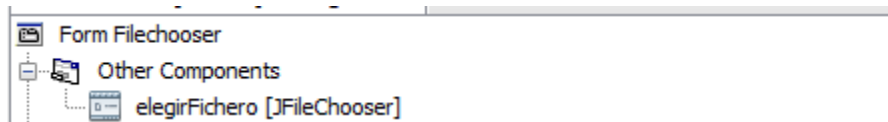
Con una barra de menú que contendrá un JMenuItem, que a la vez tendrá dos JMenuItem, de la siguiente manera.



Las cuales se cambiarán de las variables de la siguiente manera.



El JFileChooser se agrega de la parte de otros componentes.



Ya que nuestro botón Abrir será el principal en esta actividad, se debe codificar es su Evento Action.

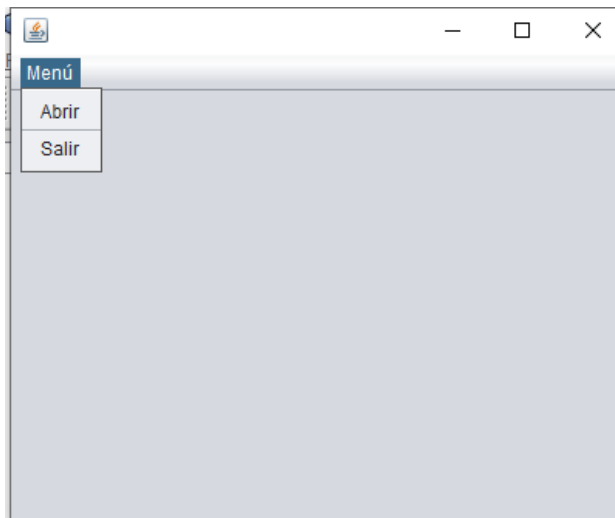
Se debe importar las siguientes librerías.

```
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
```

Y nuestro botón abrir lo siguiente.

```
private void menuAbrirActionPerformed(java.awt.event.ActionEvent evt) {
    int resp;
    resp=elegirFichero.showOpenDialog(this);
    if (resp==JFileChooser.APPROVE_OPTION) {
        JOptionPane.showMessageDialog(null,elegirFichero.getSelectedFile().toString());
    } else if (resp==JFileChooser.CANCEL_OPTION) {
        JOptionPane.showMessageDialog(null,"Se pulsó la opción Cancelar");
    }
    // TODO add your handling code here:
}
```

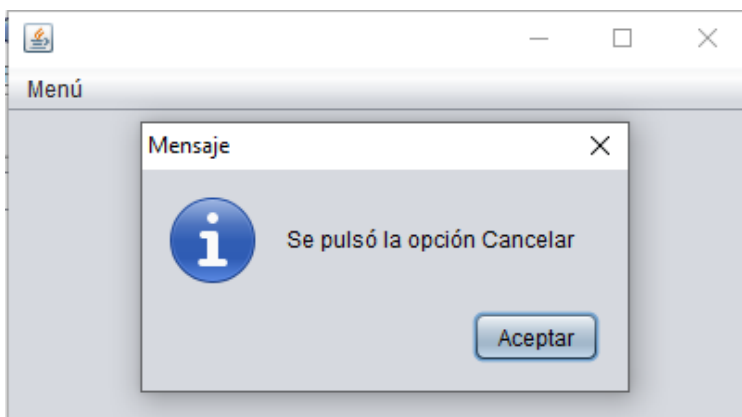
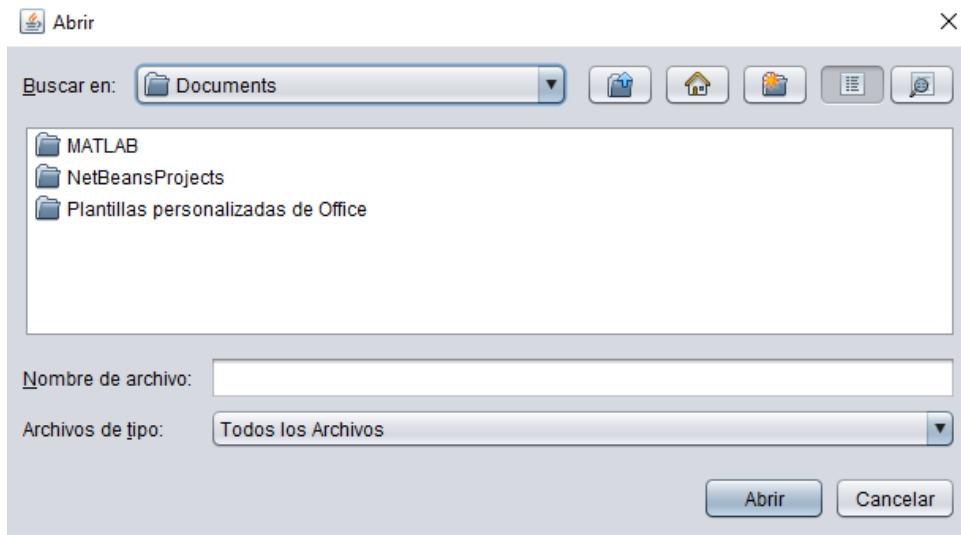
La forma en que nos aparecerá será la siguiente.



Nos saldrá la barra de menú con las dos opciones, en la cual daremos en abrir para que nos salga el gestor de archivos.

Al momento de presionar la opción abrir nos saldrá lo siguiente.

Que es nuestro Filechooser, donde podremos buscar nuestro archivo y seleccionarlo.

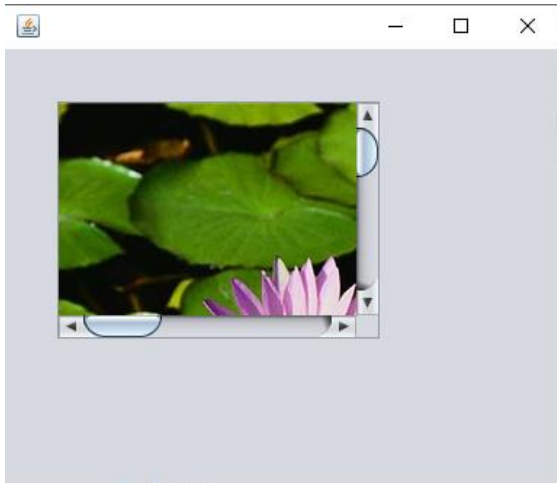


Al momento de cancelar la ventana, nos enviara lo siguiente para que estemos en cuenta que no se realizó la acción.

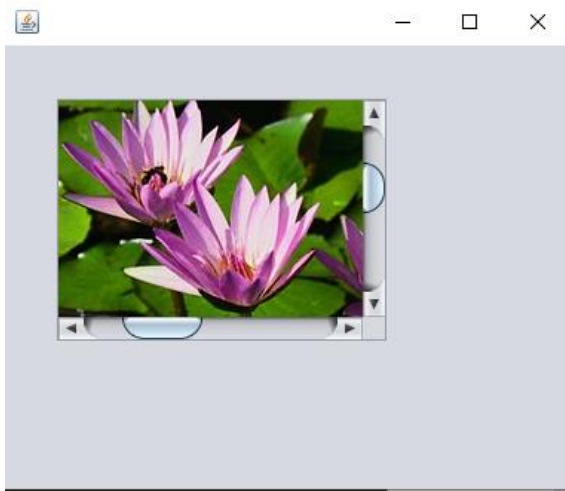
ScrollPane.

Para las dos siguientes actividades veremos el uso del JScrollPane, el cual puede ser muy útil cuando nuestro contenido es muy grande y no se puede ver completamente.

Se agregará el componente JScrollPane y dentro un de ello un JLabel el cual se cambiará su fondo con la propiedad de icon y se pondrá un fondo de imagen como a continuación.



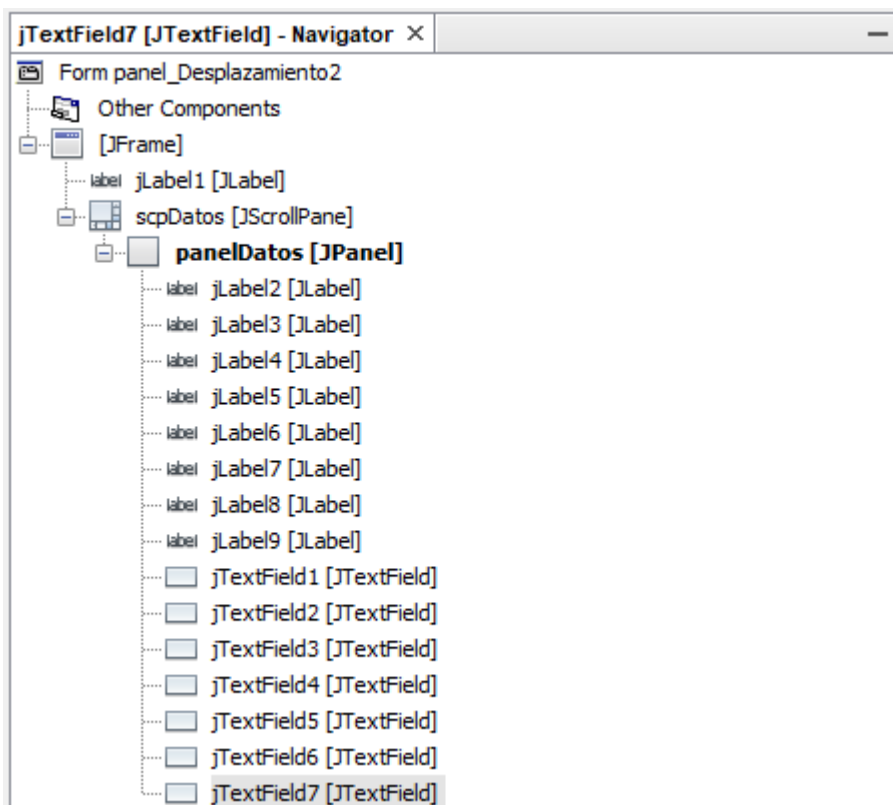
Como se puede apreciar no se ve completamente, pero con el ScrollPane se puede ver, como se muestra a continuación.



Como se ve, la imagen se puede apreciar de otra forma, ya que, con el componente, nos ayuda a ir dirigiendo hasta donde nosotros queramos.

La segunda actividad relacionada a este tema es con un panel ya que de la misma manera puede contener muchos componentes, pero si el JFrame es pequeño no se podrá ver completamente.

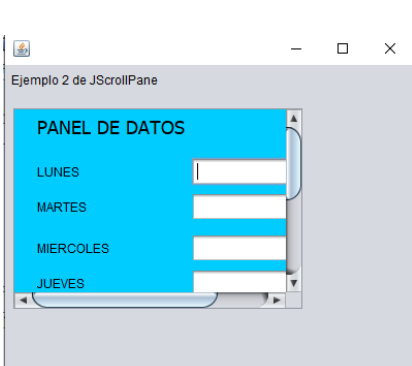
Por lo cual se agregaron los siguientes componentes.



Lo siguiente es verlo en panel que nos aparecerá completo, mientras que en el ScrollPane no.



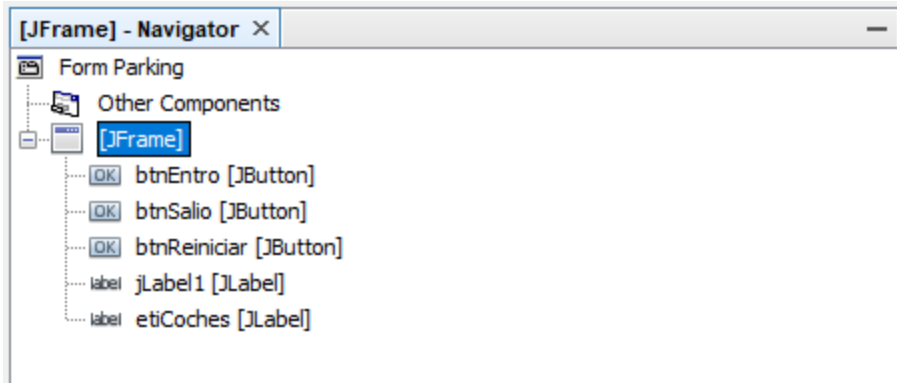
Y cuando es ejecutado de la siguiente manera.



Variables globales.

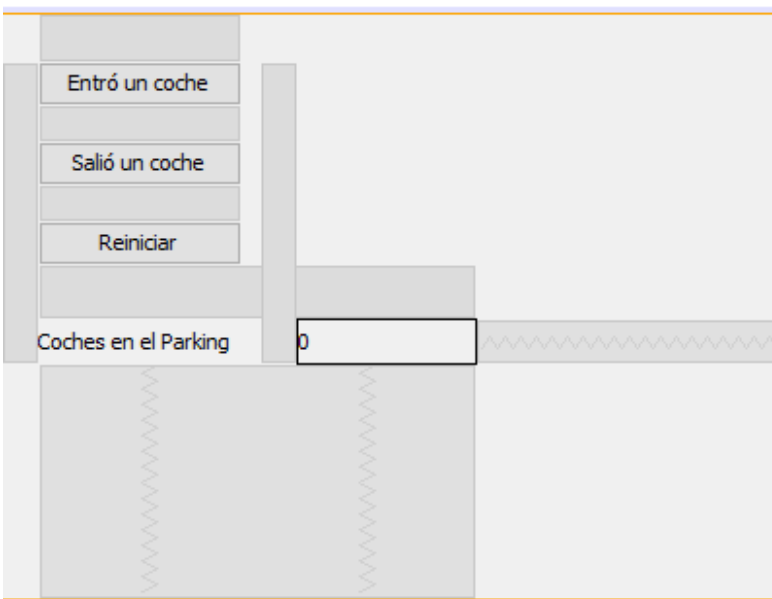
Para esta practica debemos implementar las variables globales las cuales nos ayudaran a utilizarlas en toda nuestra clase.

Primero debemos crear el proyecto, con nuestro paquete, y el JFrame.



Se les cambiara el nombre a las variables, y quedara de la siguiente manera.

Los componentes se distribuirán de la siguiente manera:



Como se puede observar, será un contador de carros donde el entrar será suma y el salir será restar, mientras que reiniciar nos devuelve a 0.

Se debe declarar la variable global de la siguiente manera.

```
public class Parking extends javax.swing.JFrame {  
    int coches;
```

Al igual que se debe inicializar en 0 en nuestro constructor, para que cada vez que se ejecute nuestro programa de la siguiente manera.

```
public Parking() {  
    initComponents();  
    coches=0;  
}
```

En nuestro botono entroCoche será un sumador por lo cual el código será el siguiente.

```
private void btnEntroActionPerformed(java.awt.event.ActionEvent evt) {  
    coches=coches+1;  
    etiCoches.setText(""+coches);  
    // TODO add your handling code here:  
}
```

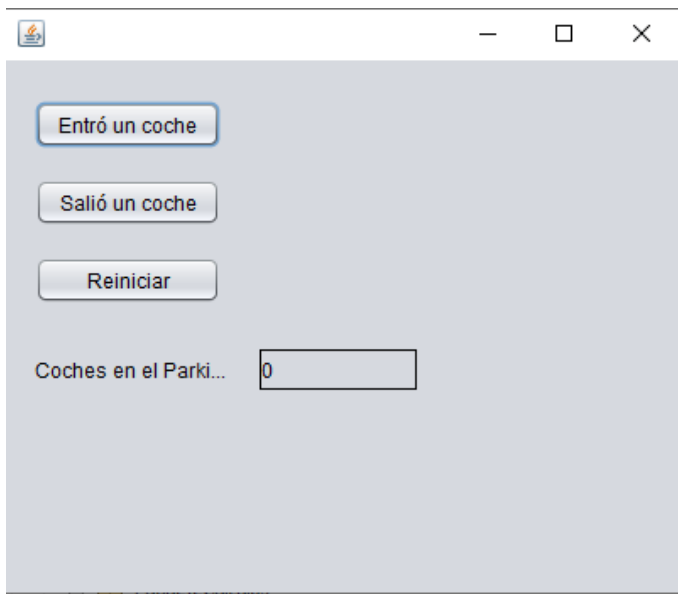
Mientras el que el botón salioCoche será un restador como se ve a continuación.

```
private void btnEntroActionPerformed(java.awt.event.ActionEvent evt) {  
    coches=coches+1;  
    etiCoches.setText(""+coches);  
    // TODO add your handling code here:  
}
```

Y botón Reiniciar solo volverá en cero la acción.

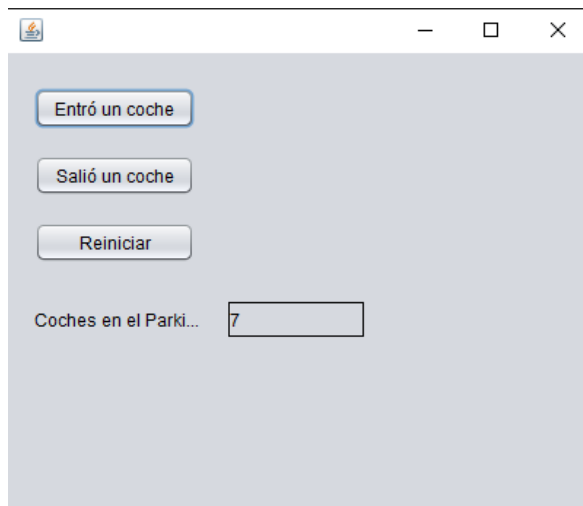
```
private void btnReiniciarActionPerformed(java.awt.event.ActionEvent evt) {  
    coches=0;  
    etiCoches.setText("0");  
    // TODO add your handling code here:  
}
```

Nuestro programa deberá verse así una vez ejecutado.



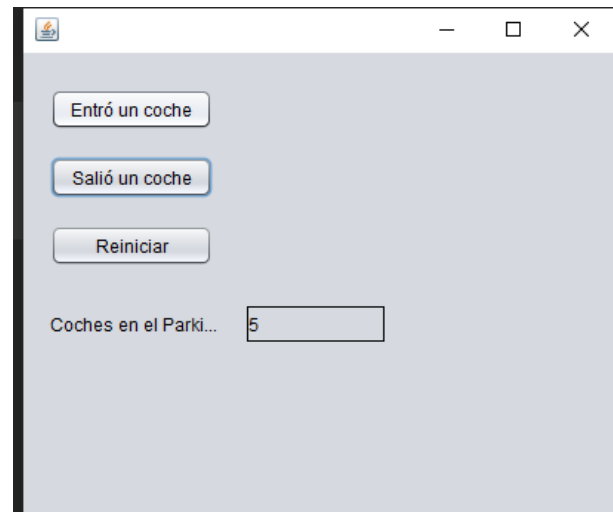
Donde nos parecerá las opciones que tenemos para poder ingresar las salidas y entradas de nuestros coches.

Sumando.



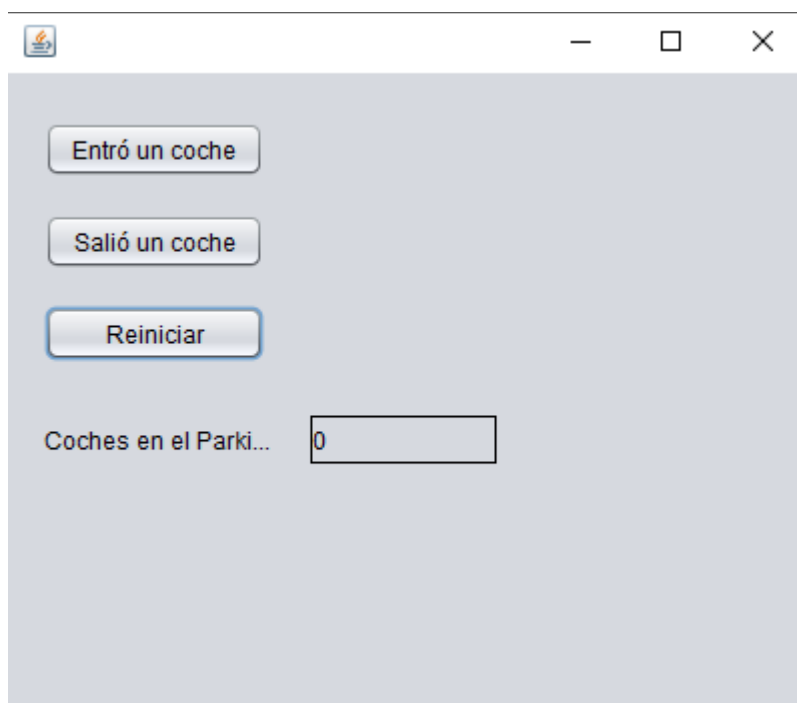
A Java Swing window titled "Sumando." with a light gray background. It contains three buttons: "Entró un coche", "Salió un coche", and "Reiniciar". Below the buttons is a text label "Coches en el Parki..." followed by a text input field containing the number "7".

Restando.



A Java Swing window titled "Restando." with a light gray background. It contains three buttons: "Entró un coche", "Salió un coche", and "Reiniciar". Below the buttons is a text label "Coches en el Parki..." followed by a text input field containing the number "5".

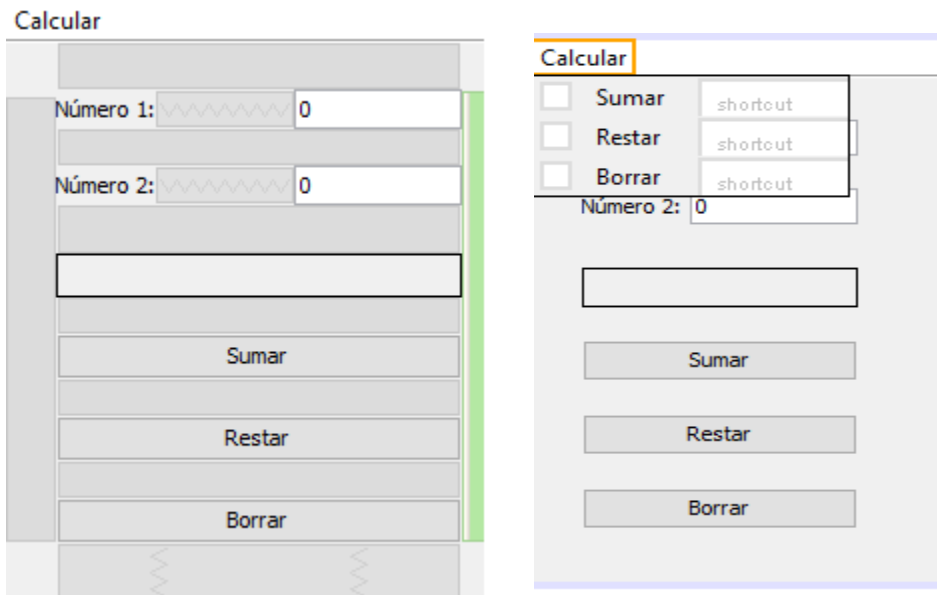
Y al momento de reiniciar nos deberá poner un cero en nuestro contador.



A Java Swing window with a light gray background. It contains three buttons: "Entró un coche", "Salió un coche", and "Reiniciar". The "Reiniciar" button is highlighted with a blue border. Below the buttons is a text label "Coches en el Parki..." followed by a text input field containing the number "0".

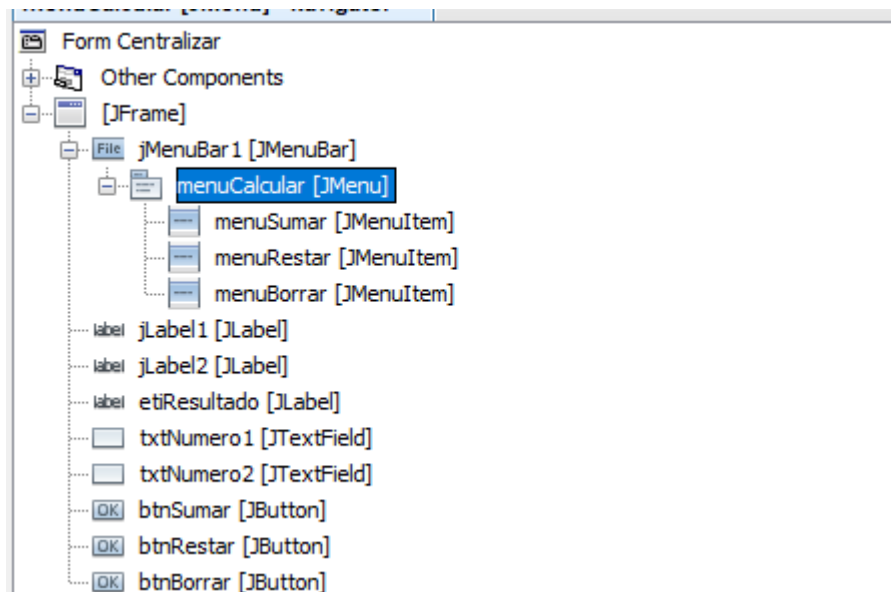
Centralizar código.

En esta actividad se implementará el uso de botones y menús por lo cual, primero debemos agregar los componentes, los cuales quedaran de la siguiente manera.



Los botones y las secciones del menú tienen el mismo funcionamiento, por lo cual no hay mucha diferencia.

Se cambiaron los nombres de las variables, lo cual quedaran de la siguiente manera.



Se tendrán que crear 3 métodos para poder realizar las operaciones, las cuales son las siguientes.

El método sumar, lo cual suma en ambos cuadros de textos.

```
void sumar() {  
    String cad1, cad2;  
    int a, b, s;  
  
    cad1 = txtNumero1.getText();  
    cad2 = txtNumero2.getText();  
    a = Integer.parseInt(cad1);  
    b = Integer.parseInt(cad2);  
    s = a + b;  
    etiResultado.setText("" + s);  
}
```

El método de restar.

```
void Restar() {  
    String cad1, cad2;  
    int a, b, r;  
  
    cad1 = txtNumero1.getText();  
    cad2 = txtNumero2.getText();  
    a = Integer.parseInt(cad1);  
    b = Integer.parseInt(cad2);  
    r = a - b;  
    etiResultado.setText("" + r);  
}
```

Y el método de borrar.

```
void Borrar() {  
    txtNumero1.setText("");  
    txtNumero2.setText("");  
}
```

Lo siguiente es ingresar los métodos en los eventos de los componentes en los cuales se ocupará.

```
private void btnBorrarActionPerformed(java.awt.event.ActionEvent evt) {  
    Borrar();  
    // TODO add your handling code here:  
}
```

```
private void menuSumarActionPerformed(java.awt.event.ActionEvent evt) {  
    sumar();  
    // TODO add your handling code here:  
}
```

```
private void btnSumarActionPerformed(java.awt.event.ActionEvent evt) {  
    sumar();  
    // TODO add your handling code here:  
}
```

```
private void txtNumero1ActionPerformed(java.awt.event.ActionEvent evt) {
    sumar();

```

TODO add your handling code here:

```
}

```

```
private void txtNumero2ActionPerformed(java.awt.event.ActionEvent evt) {
    sumar();

```

// TODO add your handling code here:

```
}

```

```
private void btnRestarActionPerformed(java.awt.event.ActionEvent evt) {
    Restar();

```

// TODO add your handling code here:

```
}

```

```
private void menuRestarActionPerformed(java.awt.event.ActionEvent evt) {
    Restar();

```

// TODO add your handling code here:

```
}

```

```
private void menuBorrarActionPerformed(java.awt.event.ActionEvent evt) {
    Borrar();

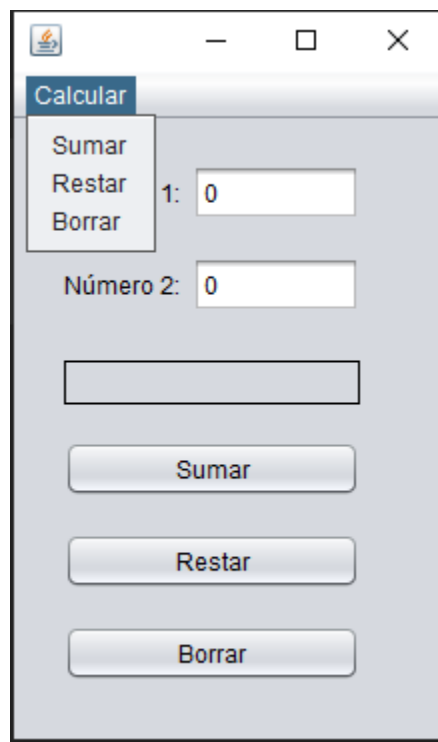
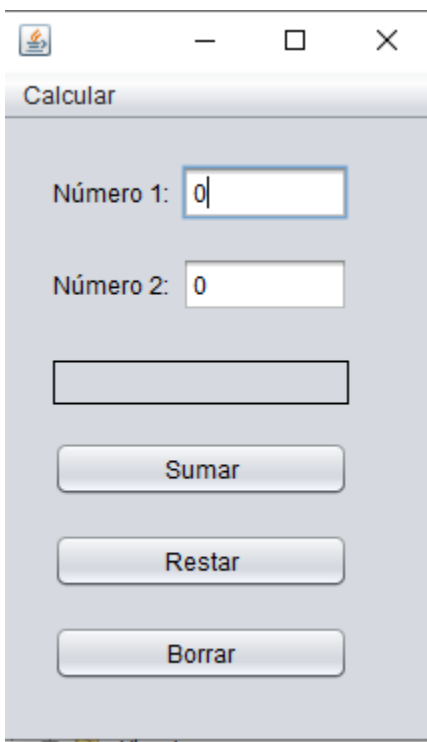
```

// TODO add your handling code here:

```
}

```

Nuestra interfaz se debe ver de la siguiente manera ya ejecutada.



Como se puede observar en el menú y con los botones de puede realizar las acciones.

Suma.

A screenshot of a 'Calculadora' window titled 'Suma.'. The window has a title bar with a small icon and standard window controls. The title bar text is 'Calculadora'. The main area contains two input fields: 'Número 1:' with the value '6' and 'Número 2:' with the value '8'. Below these is a result field containing '14'. At the bottom are three buttons: 'Sumar', 'Restar', and 'Borrar'. The 'Sumar' button is highlighted with a blue border.

Resta.

A screenshot of a 'Calculadora' window titled 'Resta.'. The window has a title bar with a small icon and standard window controls. The title bar text is 'Calculadora'. The main area contains two input fields: 'Número 1:' with the value '10' and 'Número 2:' with the value '8'. Below these is a result field containing '2'. At the bottom are three buttons: 'Sumar', 'Restar', and 'Borrar'. The 'Restar' button is highlighted with a blue border.

Borrar.

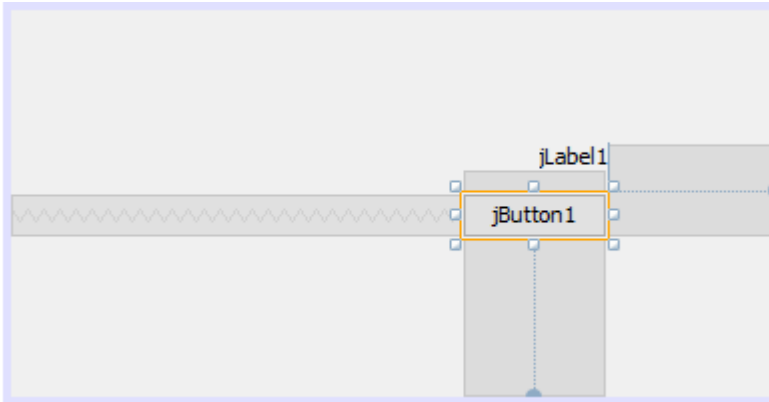
A screenshot of a 'Calculadora' window titled 'Borrar.'. The window has a title bar with a small icon and standard window controls. The title bar text is 'Calculadora'. The main area contains two input fields: 'Número 1:' with the value '10' and 'Número 2:' with the value '8'. Below these is a result field containing '2'. At the bottom are three buttons: 'Sumar', 'Restar', and 'Borrar'. The 'Borrar' button is highlighted with a blue border.

Así es como concluye esta actividad.

Layaouts.

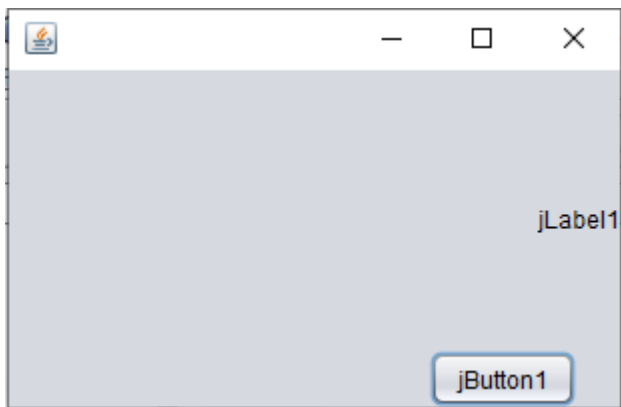
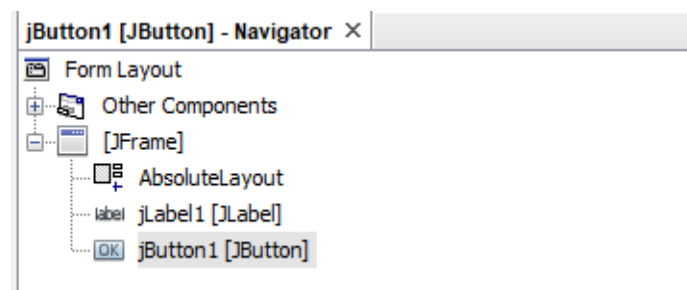
Diseño libre.

El Frame nos dará una ayuda de como colocar de una mejor manera.



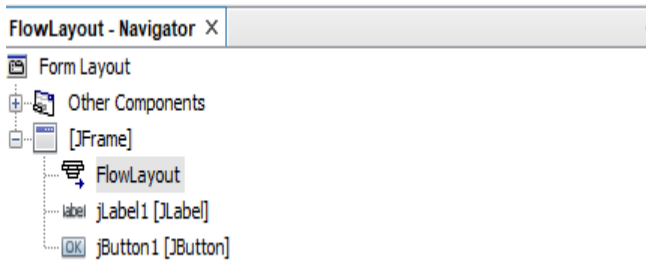
Posición Absoluta

Todo el diseño será nuestro pulso, y la colocación cuando se aumente será diferente.

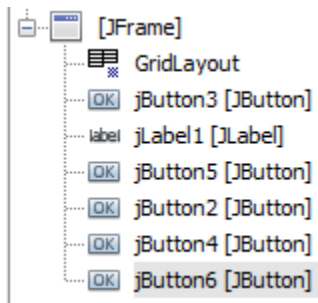


Distribución en línea.

Nos coloca un componente atrás de otro.



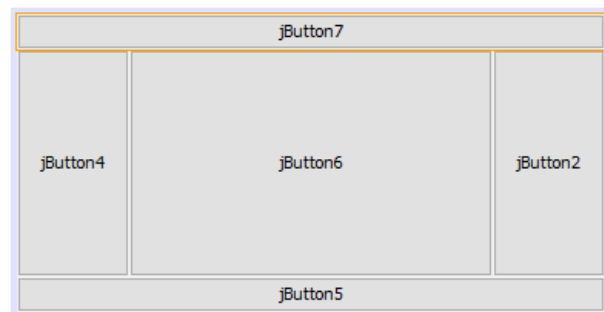
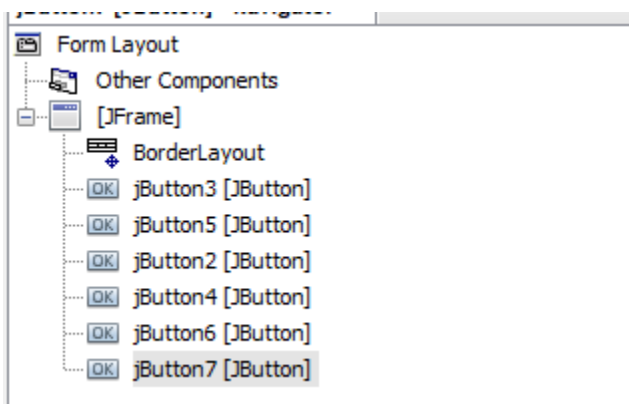
Distribución en rejilla.



Properties		
Columns	3	...
Horizontal Gap	2	...
Rows	2	...
Vertical Gap	0	...

BorderLayout

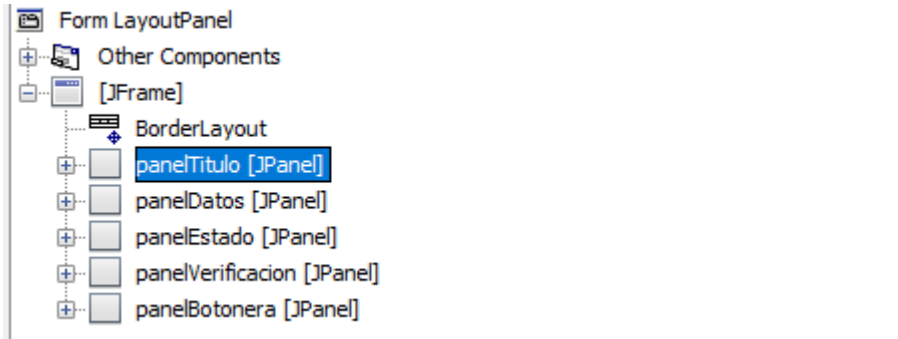
Se contendrá todo en los bordes y centros.



Layaouts y paneles.

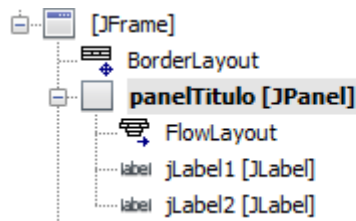
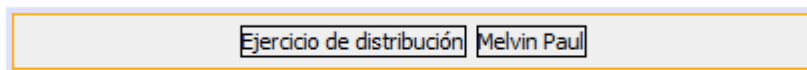
En esta actividad será una combinación de todas los componentes y distribuciones que hemos visto, por lo cual se explicara por cada panel que se hizo.

Se creo un JFrame el cual se ocupará con el BorderLayout, que a la vez se agregaran 5 paneles, distribuidos por los laterales y un central.



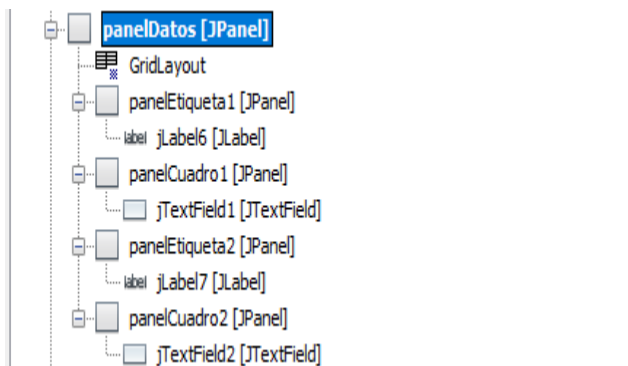
Panel Titulo.

Usan el diseño de Layaouts y con etiquetas de la siguiente manera.



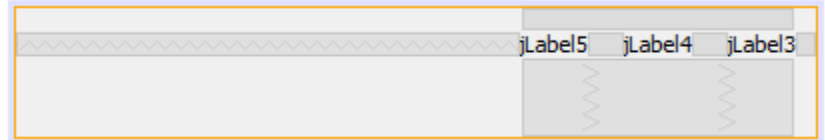
Panel de Datos.

El cual usa el GridLayout, que se divide en 4 con paneles.



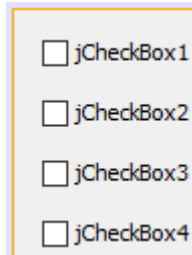
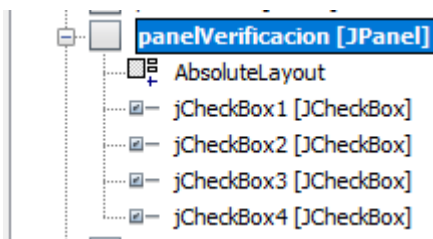
Panel de Estado.

Este será con estilo libre y con etiquetas.

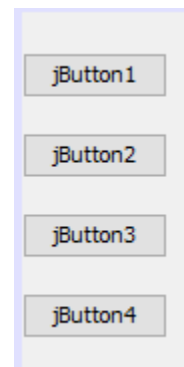
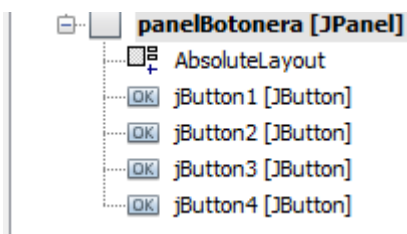


Panel de Verificación.

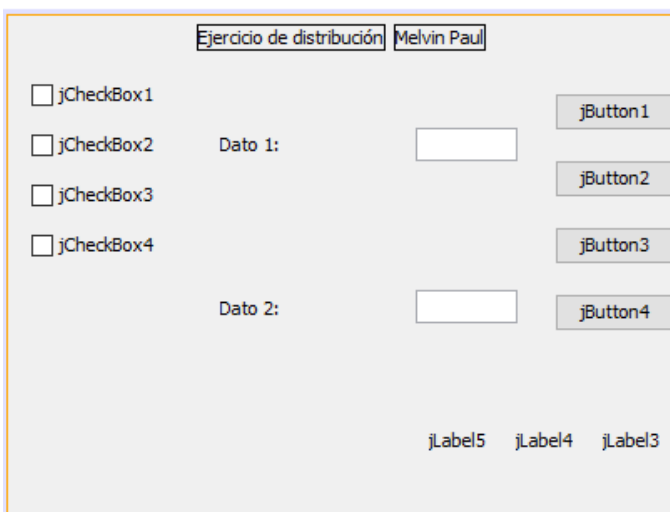
Sera con la distribución absoluta por la cual será colocados JCheckBox.



Panel Botonera.



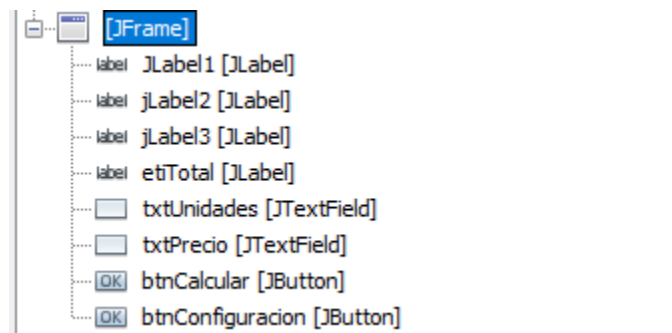
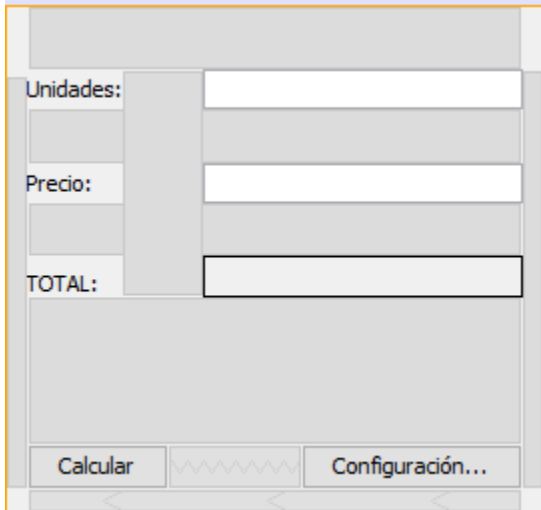
También se usa la distribución absoluta la cual queda la siguiente manera.



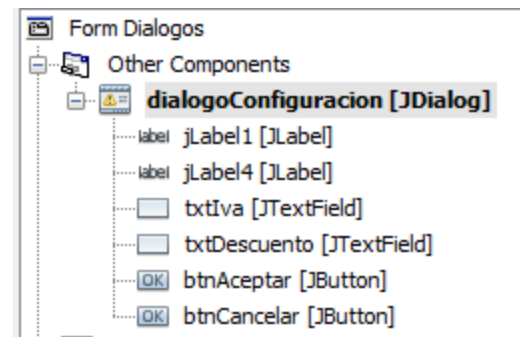
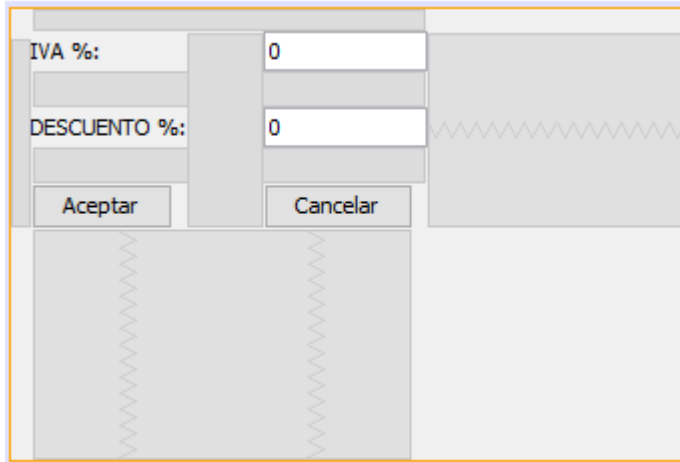
Como se puede observar, así es como queda cuando se ve por completo el JFrame y cada parte se compone por su propio panel, con sus propios componentes y distribuciones.

Diálogos.

Esta actividad usaremos el componente JDialog, el cual nos crea una nueva ventana, la cual será para modificar las acciones de una configuración, se agregarán los siguientes componentes.



Cuando agregamos JDialog se mostrará en la opción de mas componentes ya que no es visible en JFrame, al cual le agregarnos los siguientes componentes.



Para continuar con el programa, empezaremos con la creación de variables globales.

```
double iva, descuento;
```

Se debe inicializar las variables en el constructor.

```
public Dialogos() {
    initComponents();
    iva=0;
    descuento=0;
}
```

En nuestro botón calcular lo siguiente.

```
private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt) {  
    double unidades;  
    double precio;  
    double total;  
    double cantiva; //cantidad iva  
    double cantdes; //cantidad descuento  
    double totalsiniva; //total sin iva  
    //Recojo los datos de los cuadros de textos (convirtiendolos a números)  
    unidades = Double.parseDouble(txtUnidades.getText());  
    precio = Double.parseDouble(txtPrecio.getText());  
  
    //Calculo el total sin iva, la cantidad de iva y la cantidad de descuento  
    totalsiniva=precio*unidades;  
    cantiva=totalsiniva*iva/100;  
    cantdes=totalsiniva*descuento/100;  
  
    //Ahora calculo el precio total:  
    total = totalsiniva+cantiva-cantdes;  
    //Coloco el total en la etiqueta:  
    etiTotal.setText(""+total);  
    // TODO add your handling code here:  
}
```

Para que nuestro botón configuración abra la nueva ventana deberá tener lo siguiente.

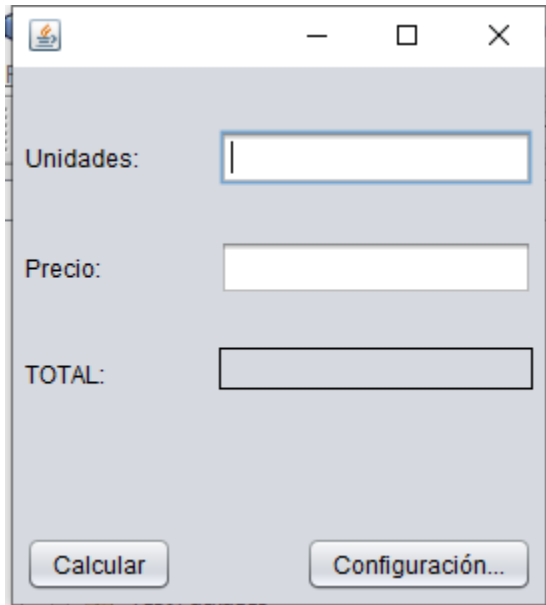
```
private void btnConfiguracionActionPerformed(java.awt.event.ActionEvent evt) {  
    dialogoConfiguracion.setSize(250,200);  
    dialogoConfiguracion.setLocation(100, 100);  
    dialogoConfiguracion.setVisible(true);  
    // TODO add your handling code here:  
}
```

Nuestro botón aceptar de JDialog lo siguiente.

```
private void btnAceptarActionPerformed(java.awt.event.ActionEvent evt) {  
    iva=Double.parseDouble(txtIva.getText());  
    descuento=Double.parseDouble(txtDescuento.getText());  
    dialogoConfiguracion.dispose();  
    // TODO add your handling code here:  
}
```

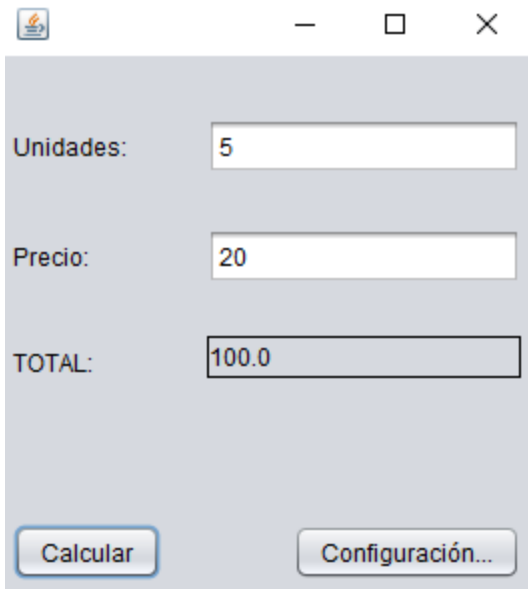
Y botón cancelar lo siguiente.

```
private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {  
    dialogoConfiguracion.dispose();  
    // TODO add your handling code here:  
}
```



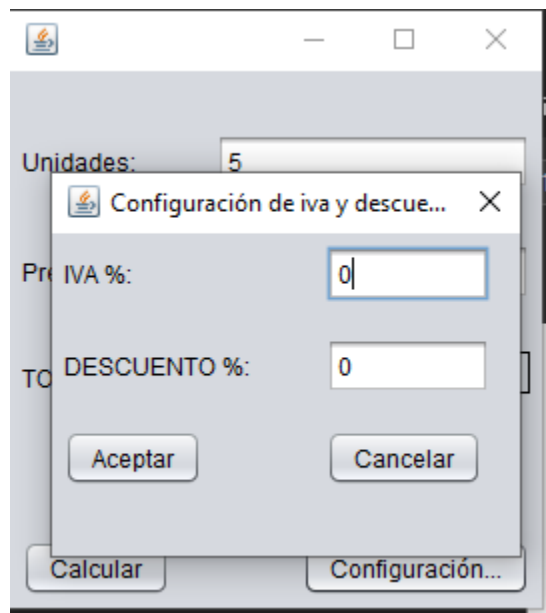
A screenshot of a Windows-style application window. The window has a title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main area is light gray and contains three labels on the left: 'Unidades:', 'Precio:', and 'TOTAL:'. To the right of each label is a white rectangular input field. The 'Unidades' field has a vertical cursor. At the bottom of the window are two buttons: 'Calcular' on the left and 'Configuración...' on the right.

Nuestra ventana principal, lo cual es sin ninguna configuración al cual como siguiente le haremos unos cálculos.

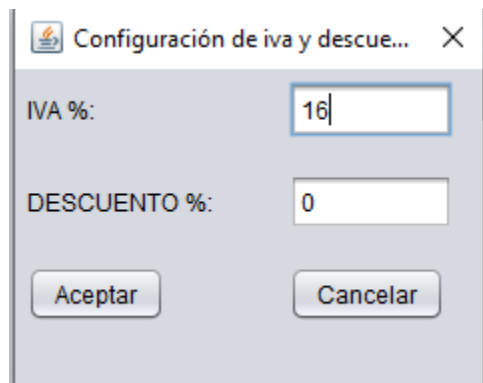


A screenshot of the same application window after a calculation. The 'Unidades' field now contains the number '5', the 'Precio' field contains '20', and the 'TOTAL' field contains '100.0'. The 'Calcular' button is highlighted with a blue border, indicating it was the last active element.

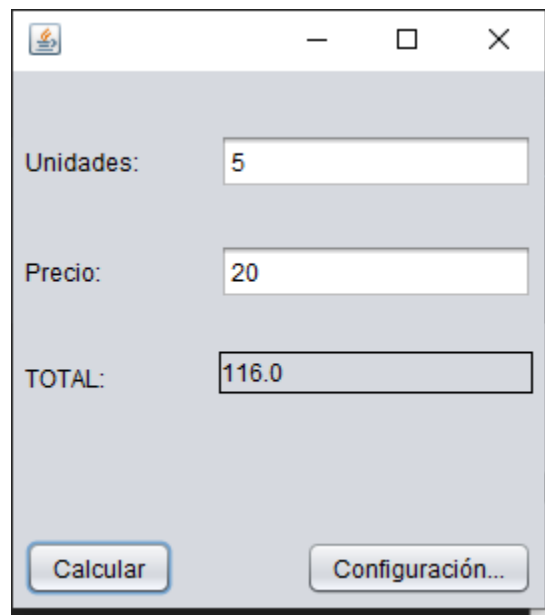
Como se puede observar se realizó con éxito el cálculo.



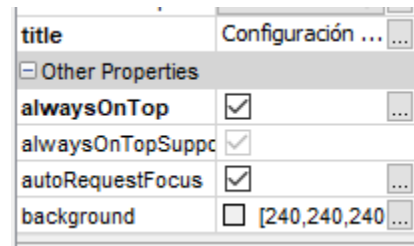
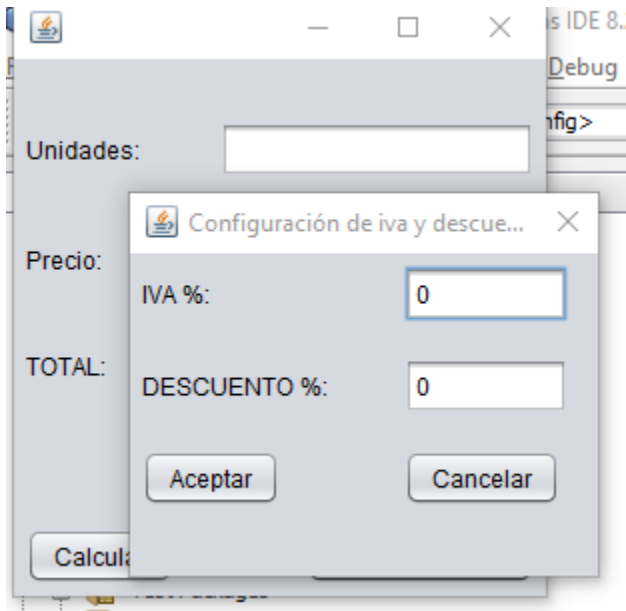
Nuestra ventana que nos ayudara configurar, en segundo plano.



Dando el porcentaje del IVA



Como se puede observar se calculo de la manera correcta.



Tenemos una propiedad que impida que se vayan a la ventana principal, primero deberán interactuar con la ventana de configuración.

Con esto concluimos la actividad.