

EJERCICIO GUIADO. JAVA: DIÁLOGOS

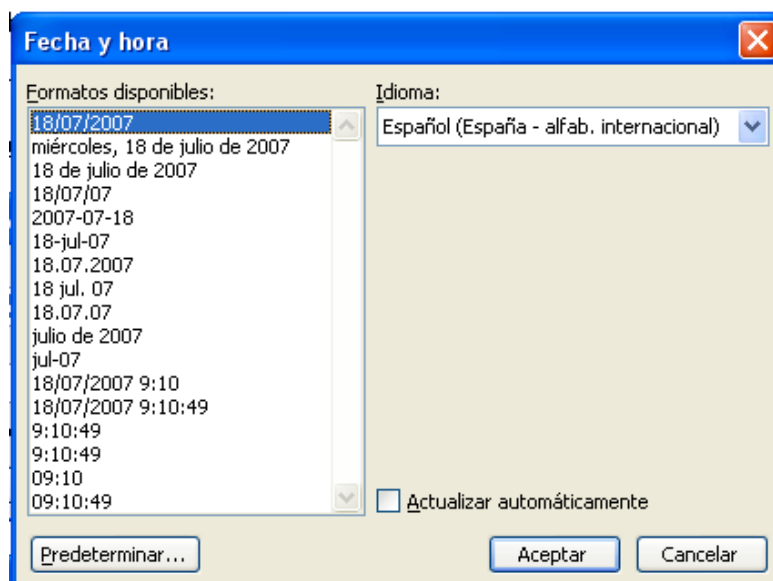
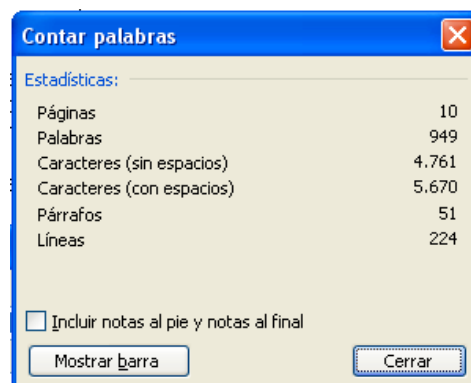
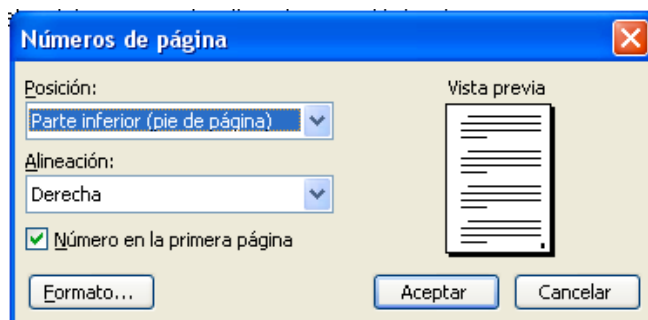
Cuadros de Diálogo

Un cuadro de diálogo es un cuadro con opciones que aparece normalmente cuando se activa una opción del menú principal del programa.

Los cuadros de diálogo tienen forma de ventana aunque no poseen algunas características de estas. Por ejemplo, no pueden ser minimizados ni maximizados.

Los cuadros de diálogo, aparte de las opciones que muestran, suelen contener dos botones típicos: el botón Aceptar y el botón Cancelar. El primero de ellos da por válidas las opciones elegidas y cierra el cuadro de diálogo. El segundo simplemente cierra el cuadro de diálogo sin hacer ninguna modificación.

He aquí algunos ejemplos de cuadros de diálogo del programa Word:



Para crear cuadros de diálogo en Java, se usa un tipo de objetos llamado JDialog. Estos objetos pueden ser diseñados como si fueran ventanas, aunque representan realmente cuadros de diálogo.

Ejercicio guiado

1. Crea un nuevo proyecto en java.
2. Diseña el JFrame de forma que la ventana tenga el siguiente aspecto:

Diagrama de una ventana de Java Swing. La ventana tiene un fondo beige y una bordura naranja. Contiene tres campos de texto con bordes azules: 'Unidades:' con un campo rectangular, 'Precio:' con un campo rectangular, y 'TOTAL:' con un campo rectangular. En la parte inferior, hay dos botones con bordes azules: 'Calcular' y 'Configuración...'. El campo 'TOTAL:' está vacío, mientras que los otros dos campos tienen un fondo blanco.

Los elementos de la ventana tendrán los siguientes nombres:

- Cuadro de texto de unidades: txtUnidades.
 - Cuadro de texto de precio: txtPrecio.
 - Etiqueta con borde del total: etiTotal.
 - Botón Calcular: btnCalcular.
 - Botón Configuración: btnConfiguracion.
3. Se pretende que cuando se pulse el botón Calcular se calcule el total de la venta (esto se hará luego) Para hacer el cálculo se tendrán en cuenta el IVA y el Descuento a aplicar. Estos dos valores serán variables globales, ya que se usarán en distintos lugares del programa.
 4. Así pues entra en el código y declara una variable global *iva* y otra *descuento* tal como se indica a continuación (recuerda que las variables globales se colocan justo después de la línea donde se define la clase principal *public class*):

```

|  * @author didact
|  */
|  public class ventanaprincipal extends javax.swing.JFrame {
|
|      double iva;
|      double descuento;
|
|  /** Creates new form ventanaprincipal */
|  public ventanaprincipal() {
|      initComponents();
|  }
  
```

Variables globales

5. Cuando el programa arranque, interesará que el *iva* por defecto sea 0, y que el *descuento* por defecto sea 0 también, así que en el constructor, inicializaremos las variables globales *iva* y *descuento* a 0:

```

-  */
|  public class ventanaprincipal extends javax.swing.JFrame {
|
|      double iva;
|      double descuento;
|
|  /** Creates new form ventanaprincipal */
|  public ventanaprincipal() {
|      initComponents();
|      iva=0;
|      descuento=0;
|  }
  
```

Inicialización de variables globales

6. Estamos ya preparados para programar el botón `btnCalcular`. Entra en su `actionPerformed` y allí se programará la realización del cálculo de la siguiente forma:

```

double unidades;
double precio;
double total;    //total
double cantiva;  //cantidad iva
double cantdes;  //cantidad descuento
double totalsiniva; //total sin iva

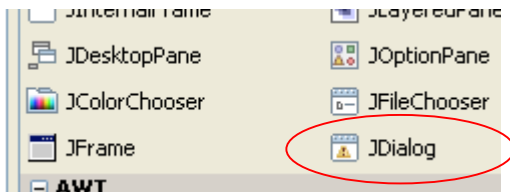
//Recojo los datos de los cuadros de textos (convirtiendolos a números)
unidades = Double.parseDouble(txtUnidades.getText());
precio = Double.parseDouble(txtPrecio.getText());

//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;
  
```

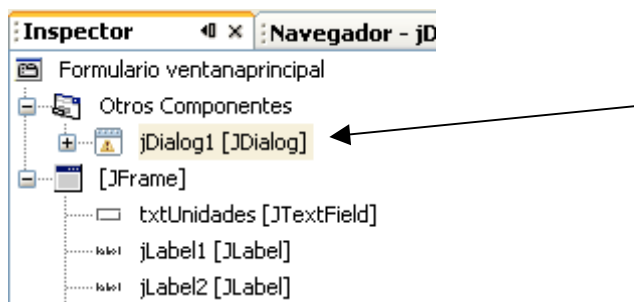
```
//Ahora calculo el precio total:
total = totalsiniva+cantiva-cantdes;

//Coloco el total en la etiqueta:
etiTotal.setText(""+total);
```

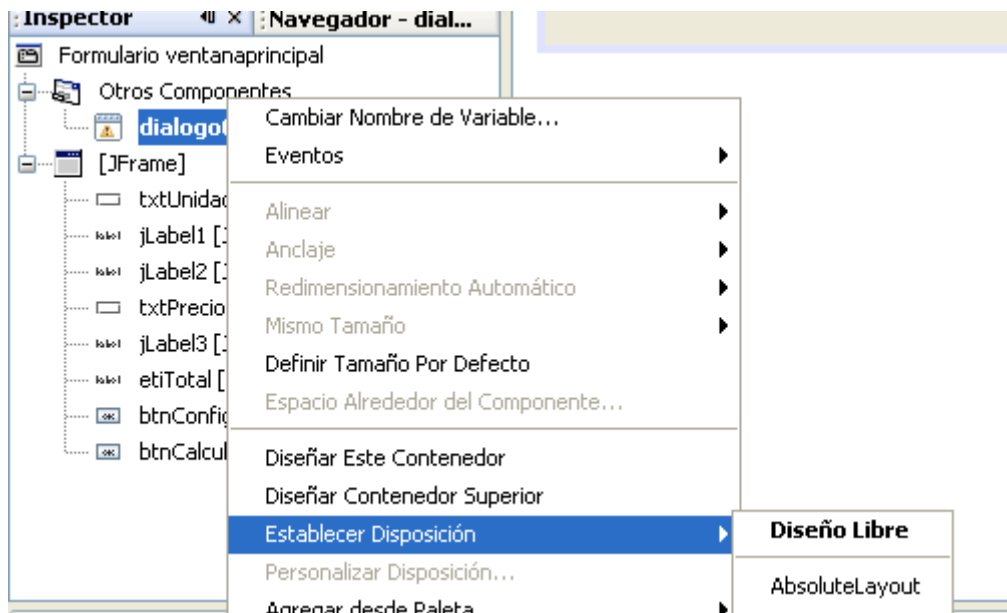
7. Puedes ya ejecutar el programa y comprobar que el botón Calcular funciona, aunque el cálculo que realiza lo hace con un iva 0 y un descuento 0.
8. A continuación se programará el botón Configuración de forma que nos permita decidir qué iva y qué descuento queremos aplicar. Este botón mostrará un CUADRO DE DIÁLOGO que permita al usuario configurar estos datos.
9. Para añadir un cuadro de diálogo al proyecto, se tiene que añadir un objeto del tipo JDialog sobre el JFrame.



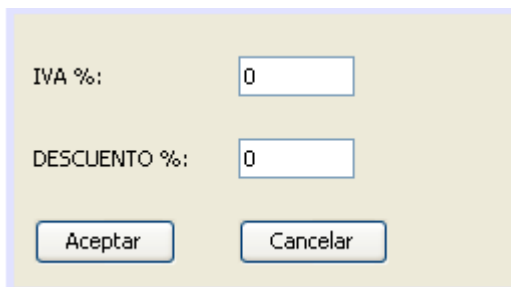
10. Los JDialog son objetos ocultos, es decir, objetos que se colocan en la parte del *Inspector* llamada *Otros Componentes*, al igual que sucede con los menús contextuales o los JFileChooser. Observa tu inspector, allí verás el JDialog que has añadido:



11. Cámbiale el nombre. Lo llamaremos *dialogoConfiguracion*.
12. Los diálogos normalmente traen por defecto el layout BorderLayout. Para nuestro ejemplo cambiaremos el layout del JDialog por el Diseño Libre:



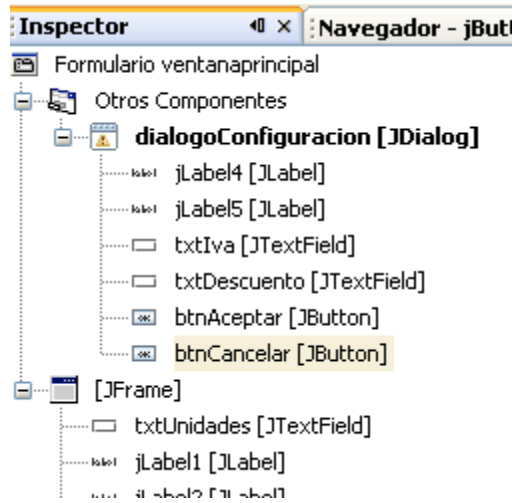
13. Los JDialog se pueden diseñar independientemente, al igual que los JPanel. Solo tienes que hacer doble clic sobre el *dialogoConfiguracion* (en el *inspector*) y este aparecerá en el centro de la ventana.
14. Así pues debes diseñar el *dialogoConfiguracion* para que quede de la siguiente forma:



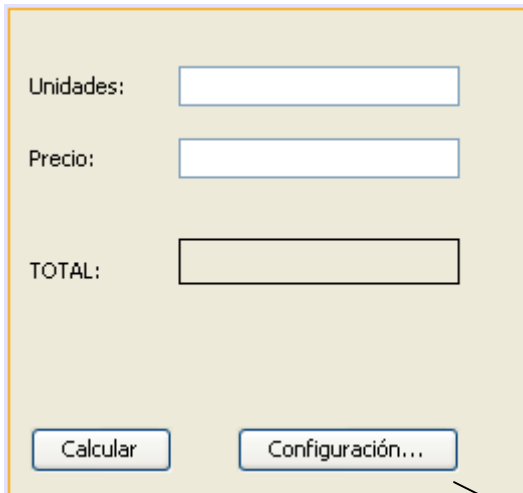
Los elementos del cuadro de diálogo tienen los siguientes nombres:

- El cuadro de texto del Iva: txtIva.
- El cuadro de texto del Descuento: txtDescuento.
- El botón Aceptar: btnAceptar.
- El botón Cancelar: btnCancelar.

Si observas el *Inspector* debe tener el siguiente aspecto:



15. Se ha dicho que cuando se pulse el botón Configuración en la ventana principal, debe aparecer el cuadro de diálogo *dialogoConfiguracion*, que acabas de diseñar:



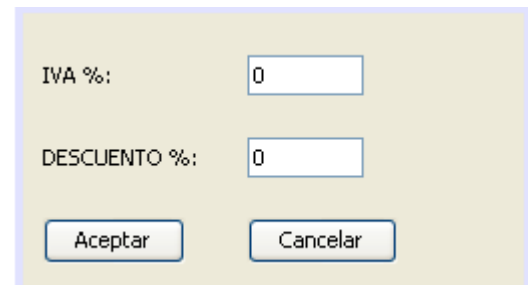
Unidades:

Precio:

TOTAL:

Calcular Configuración...

Haces clic sobre Configuración
y aparece el diálogo



IVA %:

DESCUENTO %:

Aceptar Cancelar

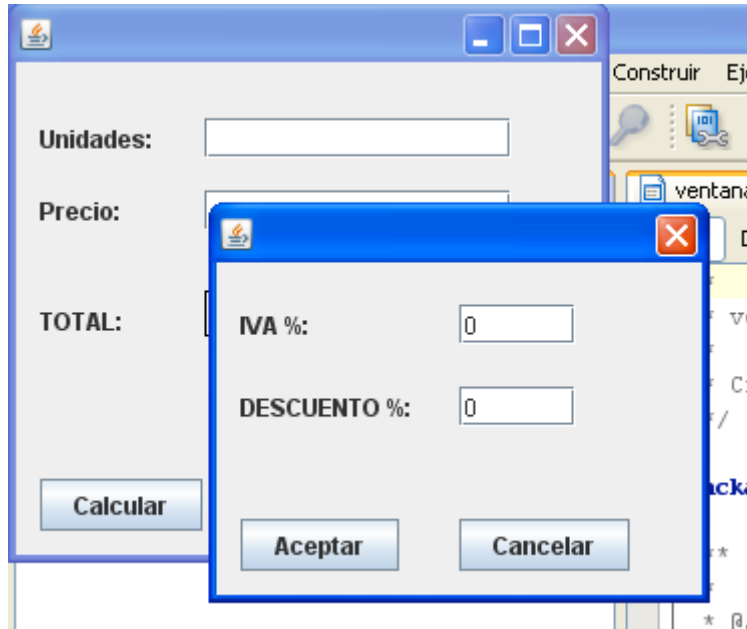
16. Para conseguir esto, debes programar el *actionPerformed* del botón *btnConfiguracion* de la siguiente forma:

```
dialogoConfiguracion.setSize(250,200);  
dialogoConfiguracion.setLocation(100,100);  
dialogoConfiguracion.setVisible(true);
```

17. El código anterior hace lo siguiente:

- A través del método *setSize* se asigna un tamaño de 250 x 200 al cuadro de diálogo.
- A través del método *setLocation* se determina que el cuadro de diálogo aparecerá en la posición (100, 100) de la pantalla.
- A través del método *setVisible* hacemos que el cuadro de diálogo se muestre.

18. Ejecuta el programa y observa lo que sucede cuando pulsas el botón Configurar. Debería aparecer el cuadro de diálogo en la posición programada y con el tamaño programado:



19. Los botones Aceptar y Cancelar del cuadro de diálogo aún no hacen nada. Así que los programaremos. Empezaremos por el más sencillo, el botón Cancelar.
20. El botón Cancelar de un cuadro de diálogo simplemente cierra dicho cuadro de diálogo. Para ello, debes añadir el siguiente código en el *actionPerformed* del botón Cancelar del diálogo:

```
dialogoConfiguracion.dispose();
```

El método *dispose* se usa para cerrar un cuadro de diálogo. También se puede usar con un *JFrame* para cerrarlo.

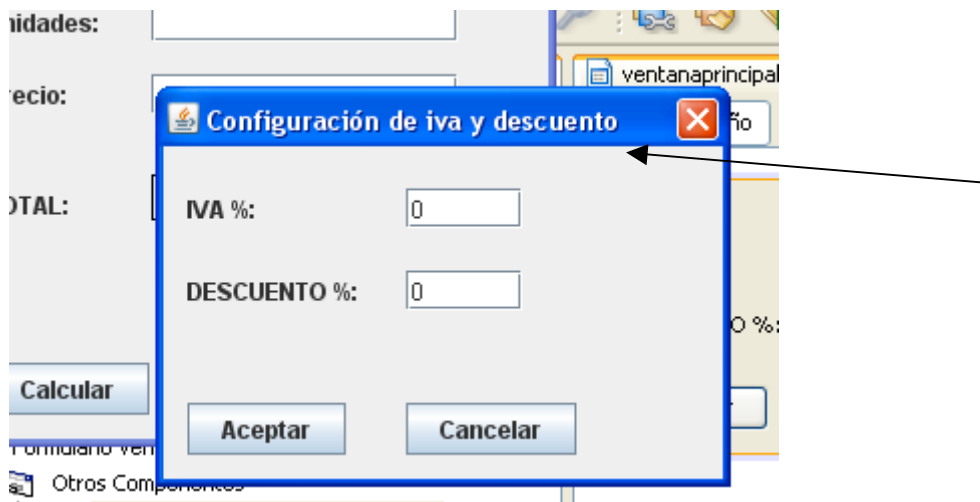
21. Ejecuta el programa de nuevo y comprueba el funcionamiento del botón Cancelar del cuadro de diálogo.
22. Ahora se programará el botón Aceptar. Cuando el usuario pulse este botón, se confirmará el valor del *iva* y del *descuento* que haya introducido. Es decir, se traspasarán los valores introducidos en los cuadros de texto *txtIva* y *txtDescuento* a las variables globales *iva* y *descuento*.

Una vez que se haya hecho esto, el cuadro de diálogo se debe cerrar.

23. Este es el código que hace lo anterior. Debe programarlo en el *actionPerformed* del botón Aceptar:


```
iva = Double.parseDouble(txtIva.getText());  
descuento=Double.parseDouble(txtDescuento.getText());  
dialogoConfiguracion.dispose();
```

24. Observe el código. Primero se traspasa los valores de los cuadros de texto a las variables globales y luego se cierra el cuadro de diálogo.
25. Compruebe el funcionamiento del programa de la siguiente forma:
 - a. Ejecute el programa.
 - b. Introduzca 5 unidades y 20 de precio.
 - c. Si pulsa calcular, el total será 100. (No hay ni iva ni descuento al empezar el programa)
 - d. Ahora pulse el botón Configuración, e introduzca un iva del 16. El descuento déjelo a 0. Acepte.
 - e. Ahora vuelva a calcular. Observe como ahora el total es 116, ya que se tiene en cuenta el iva configurado.
 - f. Pruebe a configurar un descuento y vuelva a calcular.
26. Se pretende ahora mejorar un poco el cuadro de diálogo, añadiéndole un título. Seleccione el cuadro de diálogo en el *Inspector* y luego busque su propiedad *title*. En ella escriba “*Configuración de iva y descuento*”.
27. Vuelva a ejecutar el programa. Observe la barra de título del cuadro de diálogo:

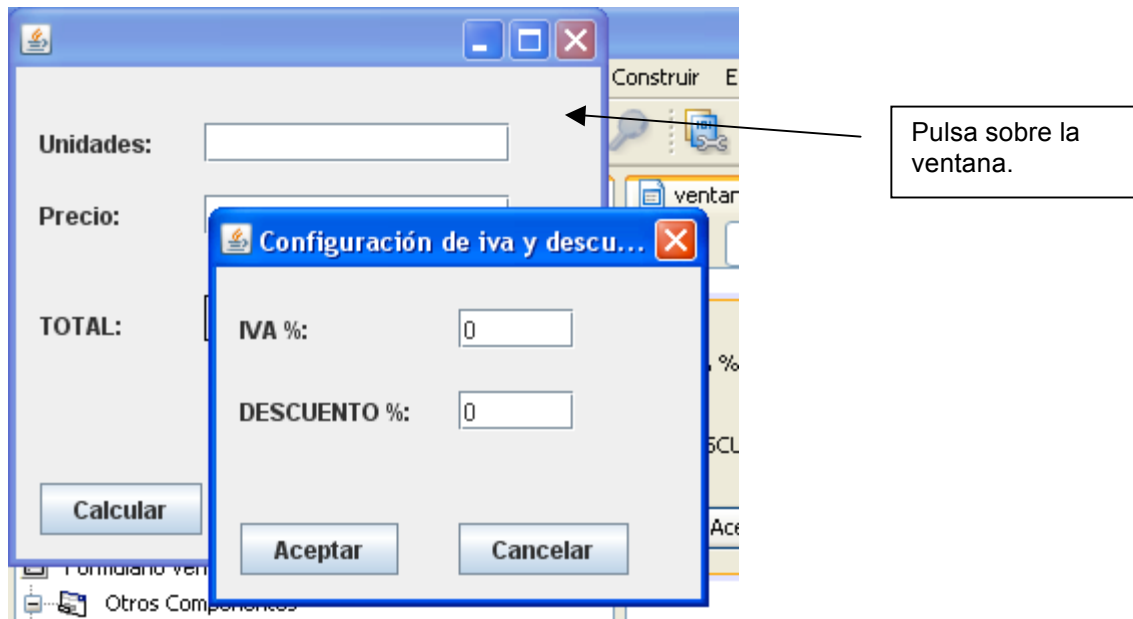


28. Ahora se estudiará el concepto de cuadro de diálogo modal y cuadro de diálogo no modal.

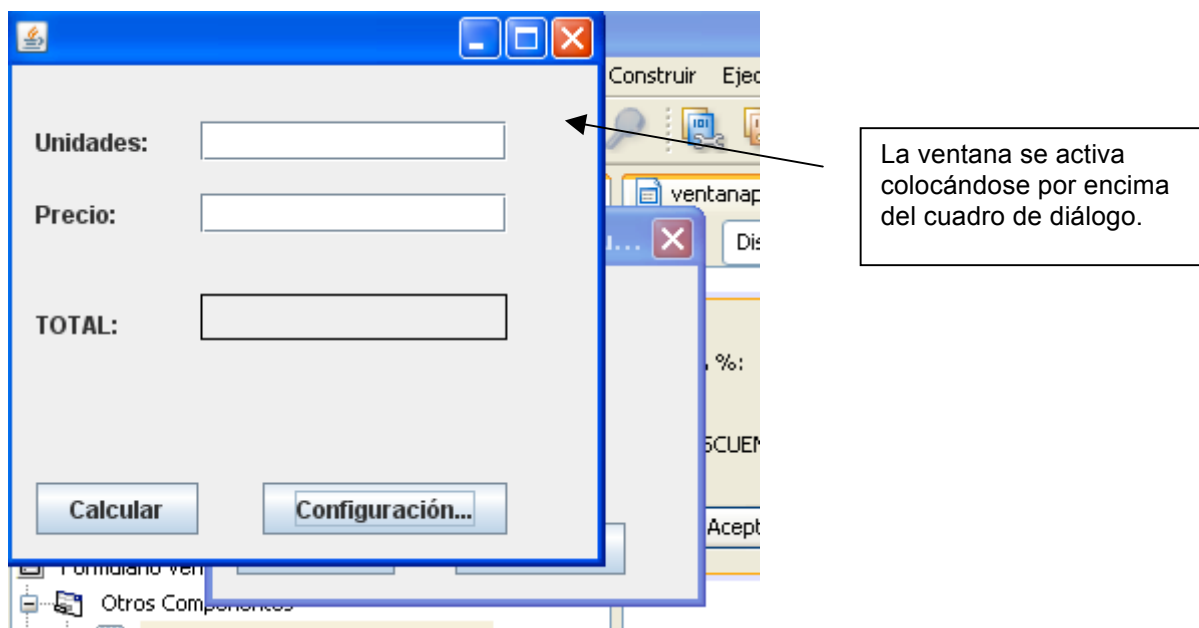
- Un cuadro de diálogo no modal. Es aquel que permite activar la ventana desde la que apareció. Los cuadros de diálogo añadidos a un proyecto son por defecto no modales.

29. Ejecuta el programa y prueba a hacer lo siguiente:
 - a. Pulsa el botón Configurar. Aparecerá el cuadro de diálogo.

b. Pulsa sobre la ventana.

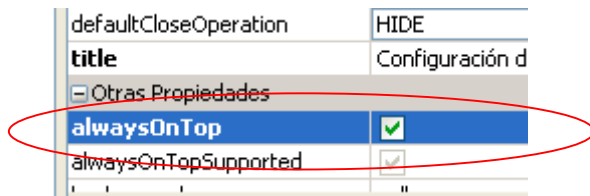


c. Observarás que la ventana se activa, colocándose sobre el cuadro de diálogo.

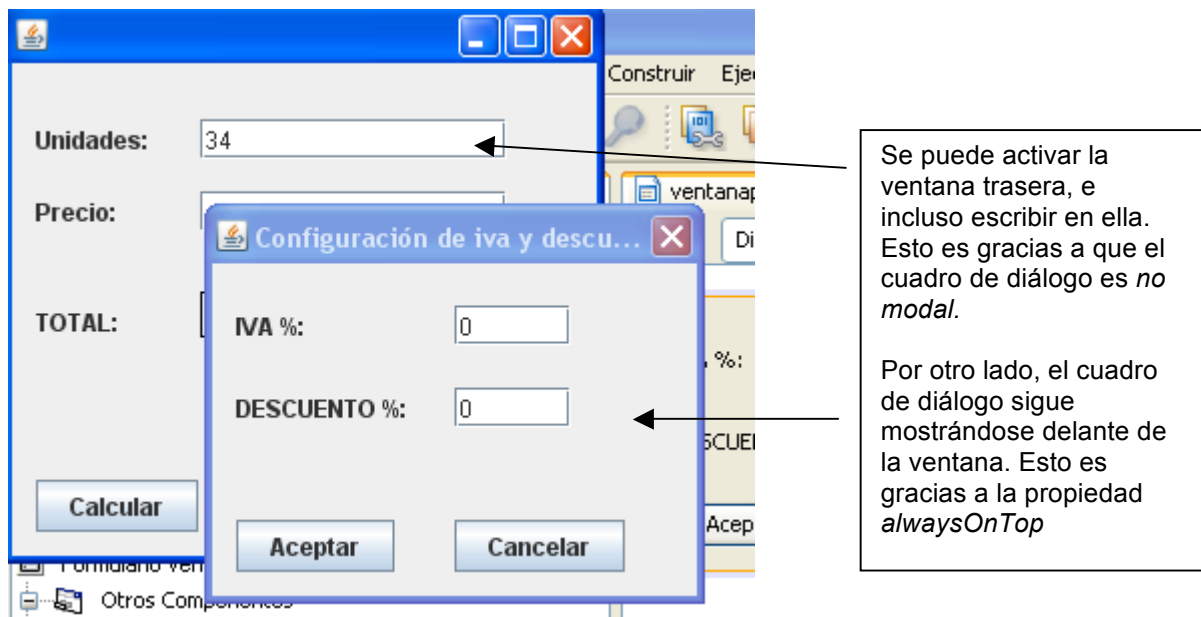


d. Esto es posible gracias a que el cuadro de diálogo es *no modal*.

e. A veces, puede ser interesante que se active la ventana pero que el cuadro de diálogo siga delante de ella. Para conseguir esto, es necesario activar la propiedad del cuadro de diálogo llamada *alwaysOnTop*. Activa esta propiedad:



- f. Ahora ejecuta el programa de nuevo y haz que se visualice el cuadro de diálogo de configuración. Podrás comprobar que se puede activar la ventana e incluso escribir en sus cuadros de textos, y que el cuadro de diálogo sigue visible:

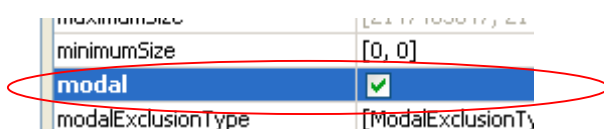


- g. Es muy común, cuando tenemos un cuadro de diálogo *no modal*, usar la propiedad *alwaysOnTop*, para que siempre aparezca delante de la ventana.

30. Ahora se estudiará el concepto de cuadro de diálogo *modal*.

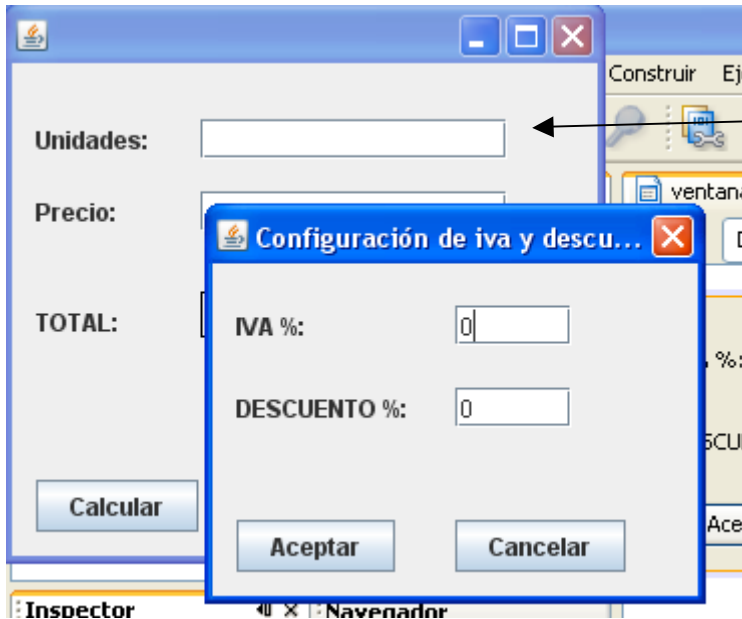
- Un cuadro de diálogo *modal* es aquel que no permite que se active otra ventana hasta que este no se haya cerrado.

31. Para convertir nuestro cuadro de diálogo en *modal*, será necesario que lo selecciones en el *inspector* y busques la propiedad *modal*. Debes activar esta propiedad.



32. Ahora ejecuta el programa comprueba lo siguiente:

- a. Haz que se visualice el cuadro de diálogo de configuración.
- b. A continuación intenta activar la ventana haciendo clic sobre ella. Verás como no es posible activarla. Es más, intenta escribir en sus cuadros de texto. No será posible hacerlo. (Incluso observarás un parpadeo en el cuadro de diálogo avisándote de ello). Esto es debido a que ahora nuestro cuadro de diálogo es *modal*.



Aunque intentes activar la ventana o escribir en ella, no podrás, ya que el cuadro de diálogo es *modal*.

Incluso verás un parpadeo en el cuadro de diálogo cuando intentas activar la otra ventana.

Se podría decir que un cuadro de diálogo *modal* es un acaparador, y que no te deja usar otro elemento hasta que no acabes con él.

Solo cuando cierres el cuadro de diálogo podrás seguir trabajando con la ventana.

- c. Solo cuando pulses, Aceptar, o Cancelar, o cierres el cuadro de diálogo, podrás seguir trabajando con tu ventana.

CONCLUSIÓN

Los Cuadros de Diálogo son ventanas simplificadas que muestran distintas opciones al usuario.

Los objetos `JDialog` son los que permiten la creación y uso de cuadros de diálogo en un proyecto java.

Para visualizar un `JDialog` será necesario llamar a su método `setVisible`. También son interesantes los métodos `setSize` para asignarles un tamaño y `setLocation` para situar el cuadro de diálogo en la pantalla.

Para cerrar un `JDialog` será necesario invocar a su método `dispose`.

Existen dos tipos de cuadros de diálogo: los modales y no modales.

Los cuadros de diálogo modales no permiten que se active otra ventana hasta que el cuadro de diálogo no se haya cerrado.

Los cuadros de diálogo no modales permiten trabajar con otra ventana a pesar de que el propio cuadro de diálogo no haya sido cerrado.