

UNIVERSIDAD DE CORDOBA

FACULTAD DE INGENIERIAS

PROGRAMA INGENIERIA DE SISTEMAS

CURSO: Programación II

TEMA: Ejemplo métodos de búsqueda en *Java*.

1. Presentación ejemplo métodos de búsqueda

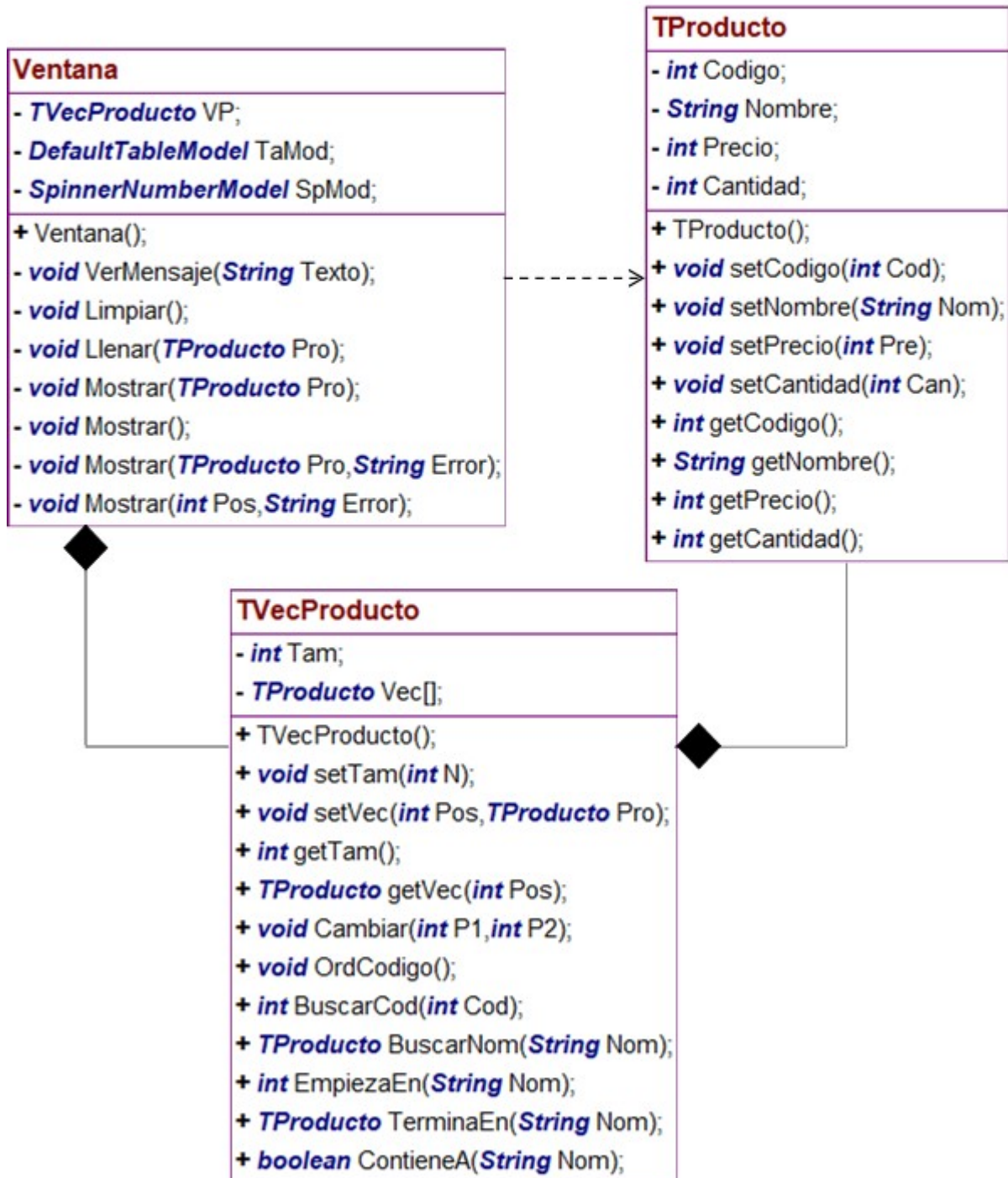
En este ejemplo se presenta la implementación de los métodos de búsqueda lineal y de búsqueda binaria, aplicados esta vez a vectores de objetos; de modo que se efectúan búsquedas sobre diferentes campos o atributos de los objetos partes almacenados en el vector. Para tal efecto, el vector almacena un conjunto de productos descritos por su código, nombre, precio y cantidad, de manera que se hace búsqueda binaria por código y búsquedas lineales por nombre; en este último caso, la clase del vector de objetos dispondrá de una búsqueda exacta por nombre y de tres búsquedas parciales por nombre así:

- ◆ Buscar productos cuyo nombre empieza por un texto dado por el usuario.
- ◆ Buscar productos cuyo nombre termina en un texto dado por el usuario.
- ◆ Buscar productos cuyo nombre contiene en un texto dado por el usuario.

Para el caso de la búsqueda binaria esta se hará por código, por ser este un valor único para cada producto; por lo tanto el arreglo se debe ordenar previamente por código y para ello emplearemos el método de ordenamiento Shell; sin embargo debe quedar claro que para este efecto cualquier método de ordenamiento es válido. Finalmente se presenta también el diagrama UML de clases de la solución para el problema –en su capa de lógica de aplicación- y se implementa en un proyecto de ventana –con *JFrame*- desarrollado en el IDE de *NetBeans*,

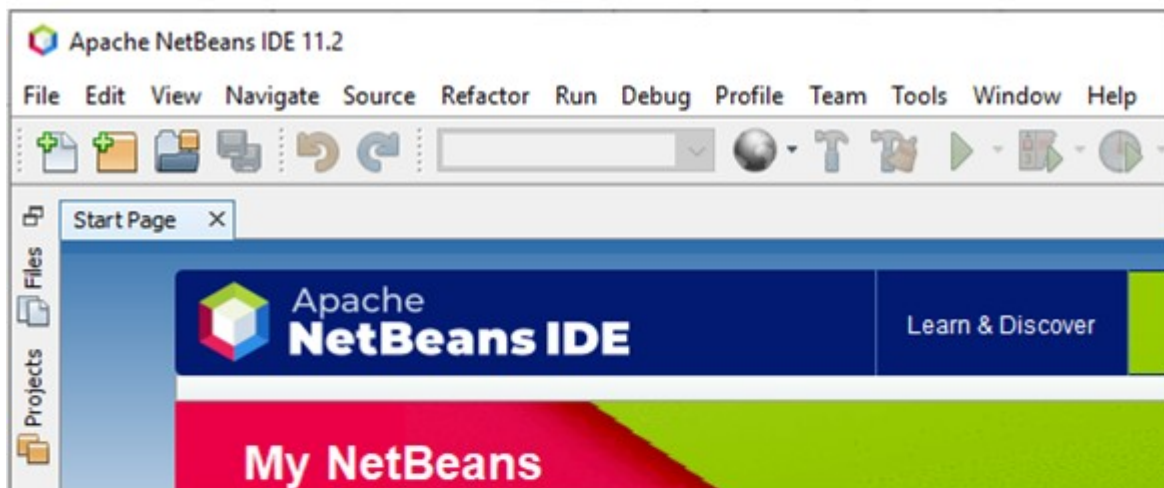
2. Diagrama UML de clases.

En la siguiente imagen de abajo observamos el diagrama de clases UML, que usaremos para implementar la solución del ejemplo planteado anteriormente.

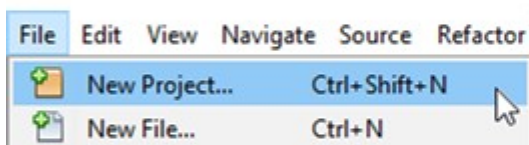


3. Creación del proyecto en *NetBeans*

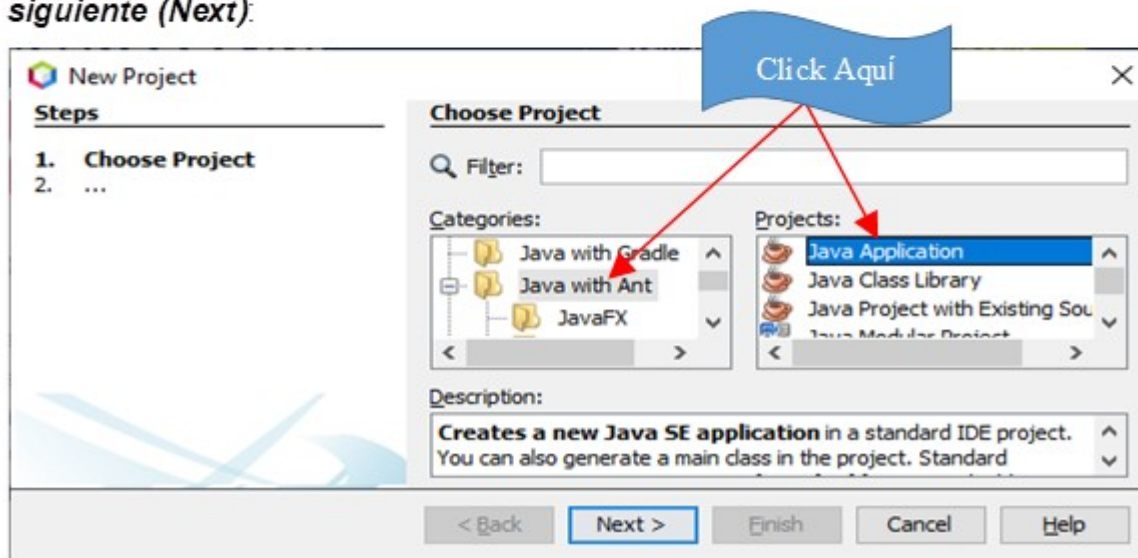
➡ Inicie o ejecute el programa *NetBeans* y verá una pantalla como la siguiente:



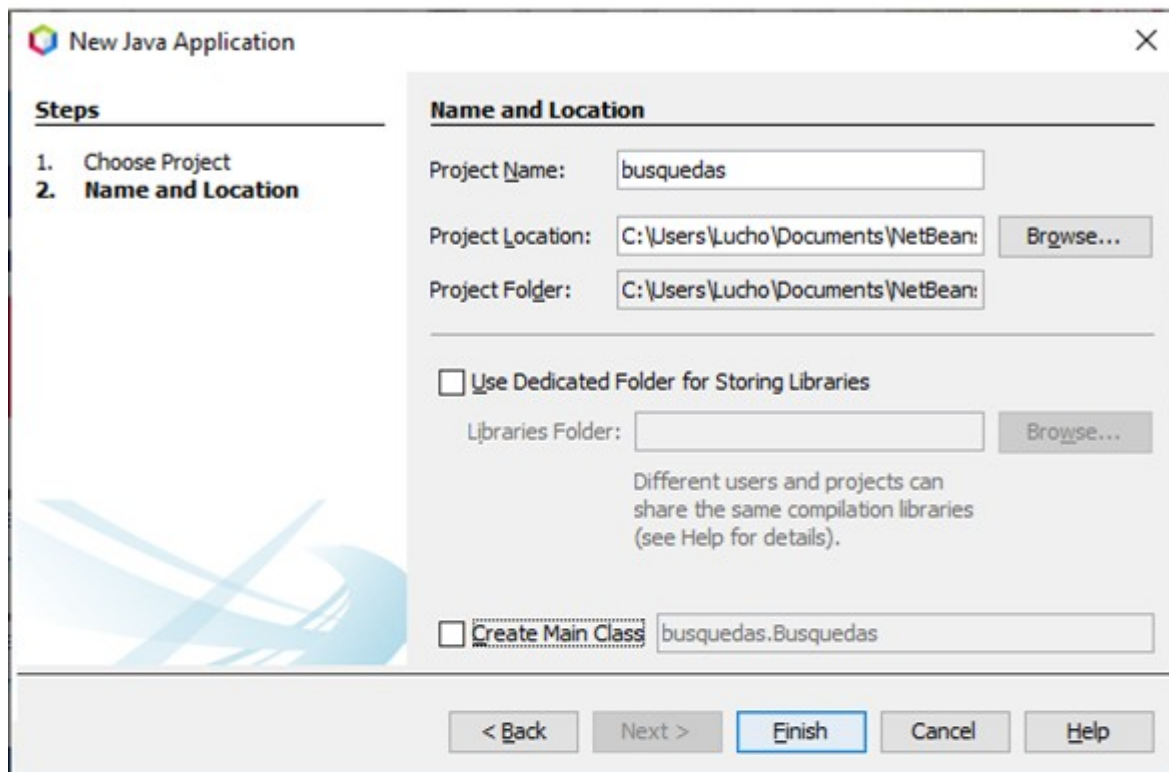
➡ Cree un nuevo proyecto haciendo click en el menú **Archivo** y luego en la opción **Proyecto nuevo**, tal como se muestra en la siguiente imagen:



➡ Ahora seleccione el nodo **Java** o **Java with Ant** del panel de **categorías** y el nodo **Java Application** del panel **proyectos**; y luego haga click en el botón **siguiente (Next)**:

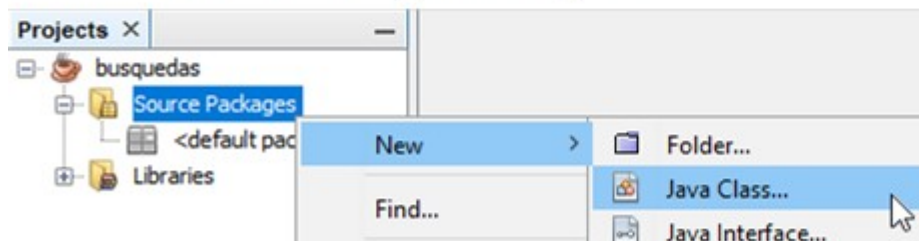


- En la siguiente ventana, ingrese el nombre del proyecto (*búsquedas*), desmarque la casilla de verificación titulada como “**Crear clase principal**” (**Create Main Class**) y haga click en el botón **Terminar** (**Finish**):

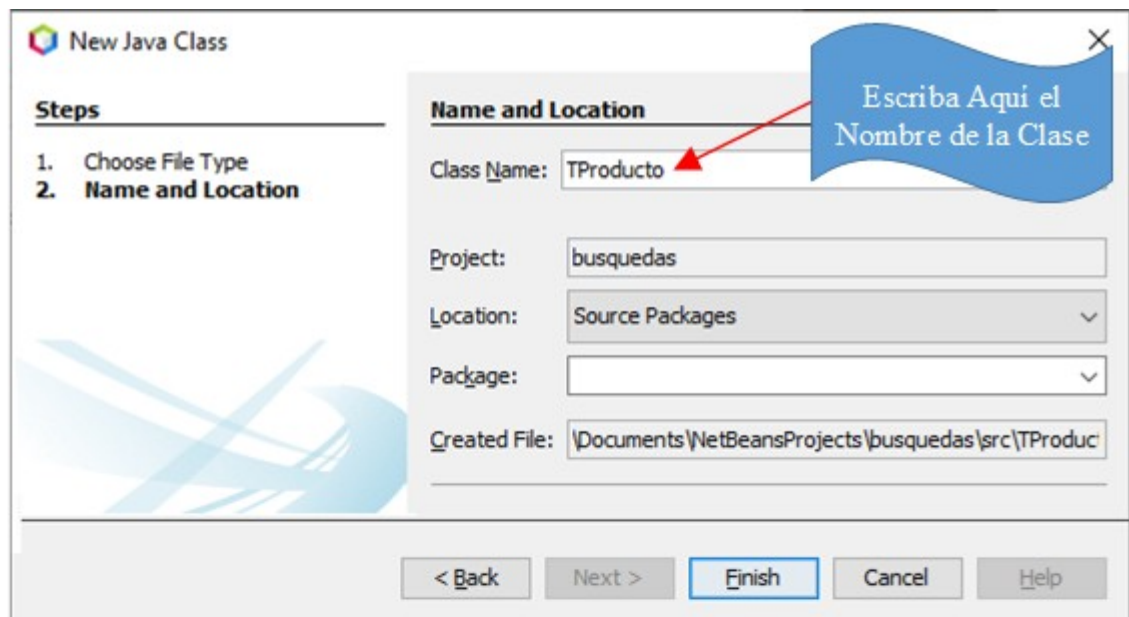


4. Implementación de la clase *TProducto*

- Para crear la clase *TProducto* haga click derecho en nodo **Source Packages** (**Paquetes fuentes**) del proyecto y escoja la opción **New** y luego **Java Class**, tal como se muestra en la siguiente imagen:



- En la ventana que se despliega, en la entrada **Class Name** ingrese el nombre para la clase (*TProducto*) y después haga click en el botón **Terminar** (**Finish**), tal como se aprecia en la siguiente imagen:



➡ Ahora implemente la clase **TProducto** de la siguiente manera:

```
TPProducto.java x
Source History
1 public class TProducto {
2
3     private intCodigo;
4     private StringNombre;
5     private intPrecio;
6     private intCantidad;
7
8     public TProducto() {
9         Codigo=0;
10        Nombre="";
11        Precio=0;
12        Cantidad=0;
13    }
14
15    public void setCodigo(int Cod) {
16        Codigo=Cod;
17    }
18
19    public void setNombre(String Nom) {
20        Nombre=Nom.trim();
21    }
22 }
```

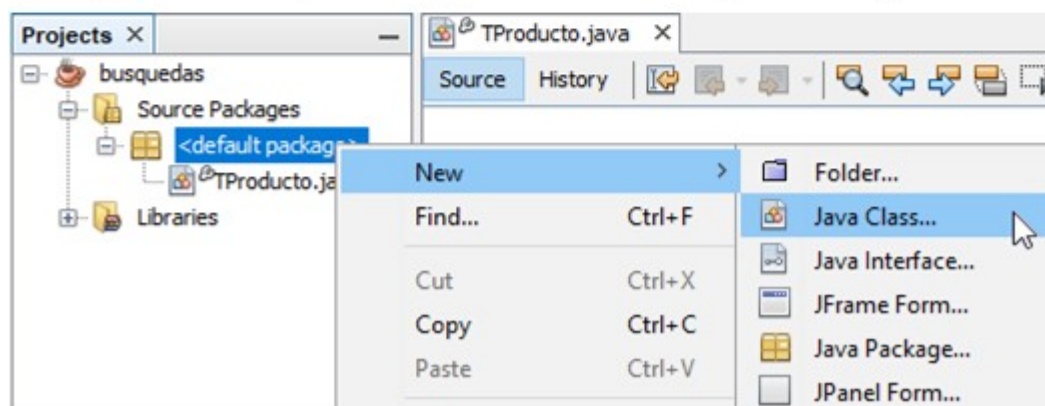
```

23  public void setPrecio(int Pre) {
24      Precio=Pre;
25  }
26
27  public void setCantidad(int Can) {
28      Cantidad=Can;
29  }
30
31  public int getCodigo() {
32      returnCodigo;
33  }
34
35  public String getNombre() {
36      returnNombre;
37  }
38
39  public int getPrecio() {
40      returnPrecio;
41  }
42
43  public int getCantidad() {
44      returnCantidad;
45  }
46
47  }

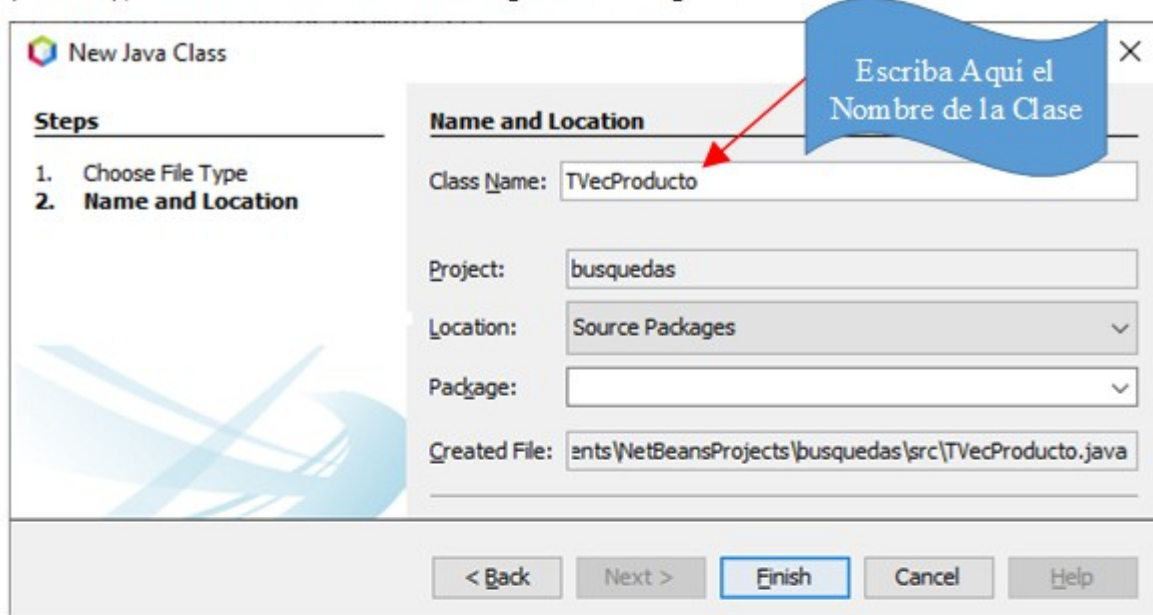
```

5. Implementación de la clase *TVecProducto*

➡ Para crear la clase ***TVecProducto*** haga click derecho en nodo **Source Packages** (**Paquetes fuentes**) del proyecto, después escoja la opción **New** y luego **Java Class**, tal como se muestra en la siguiente imagen:



- ➡ En la ventana que se despliega, en la entrada **Class Name** ingrese el nombre para la clase (**TVecProducto**) y después haga click en el botón **Terminar** (**Finish**), como se muestra en la siguiente imagen:



- ➡ Ahora implemente la clase **TVecProducto** como se muestra a continuación:

```
TVecProducto.java x
Source History
1 public class TVecProducto {
2
3     private int Tam;
4     private TProducto Vec[];
5
6     public TVecProducto() {
7         Tam=0;
8         Vec=null;
9     }
10
11     public void setTam(int N) {
12         int i;
13         Tam=N;
14         if (N>0) {
15             Vec=new TProducto[N];
16             for (i=0;i<N;i++) {
17                 Vec[i]=new TProducto();
18             }
19         }
20     }
21 }
```



```

19     }
20     else{
21         Vec=null;
22     }
23 }
24
25 public void setVec(int Pos,TProducto Pro) {
26     if(Pos>=0 && Pos<Tam) {
27         Vec[Pos]=Pro;
28     }
29 }
30
31 public int getTam() {
32     return Tam;
33 }
34
35 public TProducto getVec(int Pos) {
36     return Vec[Pos];
37 }
38
39 public void Cambiar(int P1,int P2) {
40     TProducto Pro;
41     Pro=Vec[P1];
42     Vec[P1]=Vec[P2];
43     Vec[P2]=Pro;
44 }
45
46 public void OrdCodigo() {
47     int i,j,d;
48     d=Tam/2;
49     while(d>=1) {
50         for(i=d;i<Tam;i++) {
51             j=i;
52             while(j-d>=0) {
53                 if(Vec[j-d].getCodigo()>Vec[j].getCodigo()) {
54                     Cambiar(j-d,d);
55                     j=j-d;
56                 }
57             }
58             break;
59         }

```



```

60     }
61     d=d/2;
62 }
63 }
64
65 public int BuscarCod(int Cod) {
66     int izq,der;
67     int centro,pos;
68     izq=0;
69     pos=-1;
70     der=Tam-1;
71     while(izq<=der && pos==-1) {
72         centro=(izq+der)/2;
73         if(Vec[centro].getCodigo()==Cod) {
74             pos=centro;
75         }
76         else
77             if(Cod<Vec[centro].getCodigo()) {
78                 der=centro-1;
79             }
80             else{
81                 izq=centro+1;
82             }
83     }
84     return pos;
85 }
86
87 public TProducto BuscarNom(String Nom) {
88     int i;
89     TProducto pro;
90     i=0;
91     pro=null;
92     while(i<Tam && pro==null) {
93         if(Vec[i].getNombre().equals(Nom)) {
94             pro=Vec[i];
95         }
96         else
97             i=i+1;
98     }
99     return pro;
100 }

```

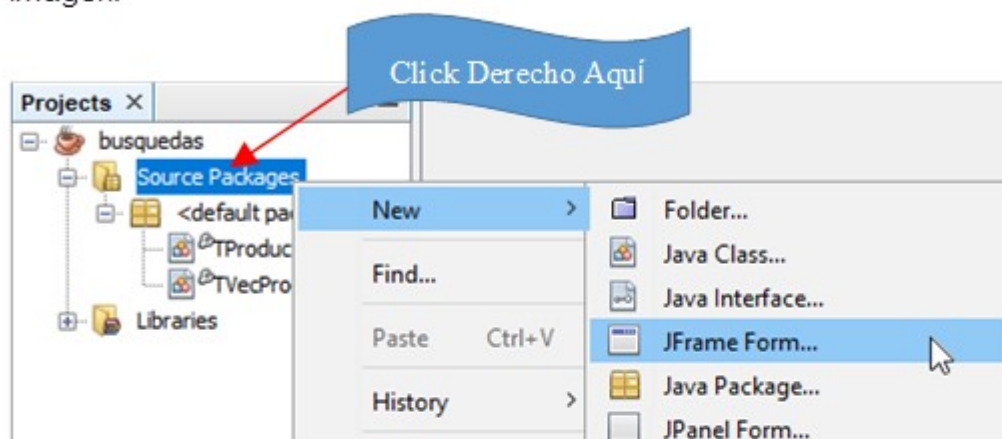
```

102  public int EmpiezaEn(String Nom) {
103      int i,pos;
104      i=0;
105      pos=-1;
106      while(i<Tam && pos==-1) {
107          if(Vec[i].getNombre().startsWith(Nom)) {
108              pos=i;
109          }
110          else
111              i=i+1;
112      }
113      return pos;
114  }
115
116  public TProducto TerminaEn(String Nom) {
117      int i;
118      TProducto pro;
119      i=0;
120      pro=null;
121      while(i<Tam && pro==null) {
122          if(Vec[i].getNombre().startsWith(Nom)) {
123              pro=Vec[i];
124          }
125          else
126              i=i+1;
127      }
128      return pro;
129  }
130
131  public boolean ContieneA(String Nom) {
132      int i;
133      boolean res;
134      res=false;
135      for(i=0;i<Tam;i++) {
136          if(Vec[i].getNombre().contains(Nom)) {
137              res=true;
138              break;
139          }
140      }
141      return res;
142  }
143  }

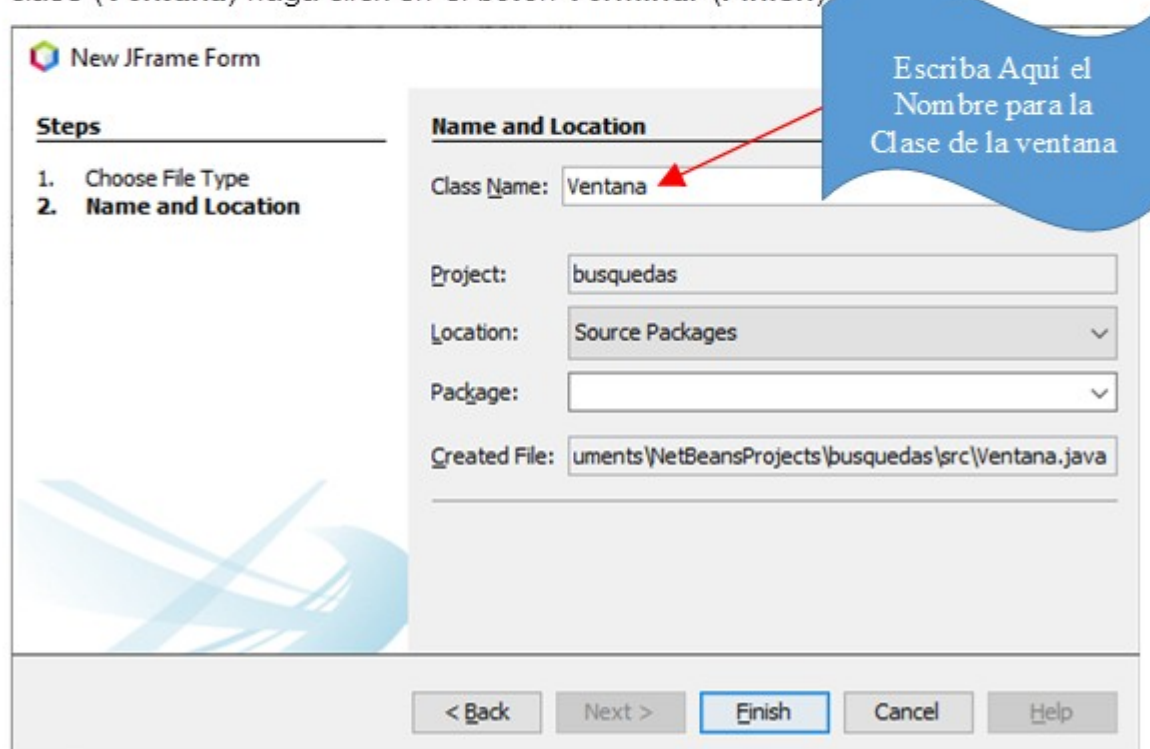
```

6. Creación de la ventana

- ➡ Cree una ventana para la interfaz gráfica de usuario, haciendo click derecho en nodo **Paquetes fuentes** del proyecto y escoja la opción **Nuevo** (o **New** si el IDE esta en inglés) y luego **Formulario JFrame**, tal como se muestra en la siguiente imagen:

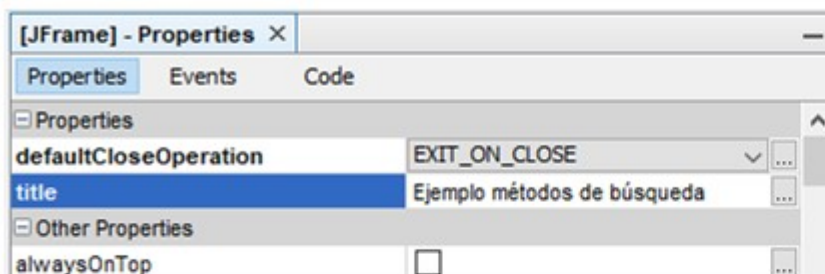


- ➡ En la ventana desplegada, en la entrada **Class Name**, ingrese el nombre para la clase (**Ventana**) haga click en el botón **Terminar (Finish)**



7. Diseño gráfico de la ventana

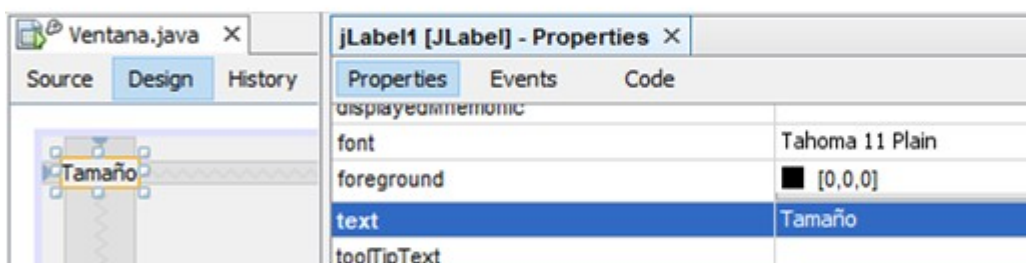
- Haga click dentro de la ventana y luego vaya al inspector de propiedades; ubique la propiedad **title** y haga click en el recuadro de su derecha; seguidamente, escriba **Ejemplo métodos de búsqueda** como título de la ventana y pulse enter.



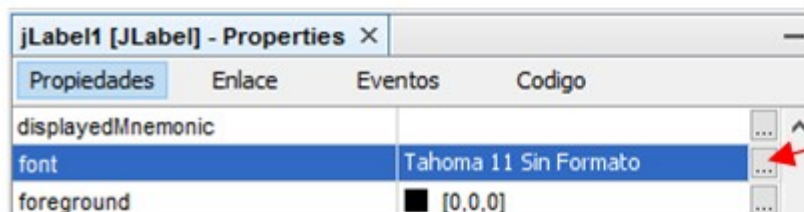
- De la paleta de controles de **Swing**, arrastre un control **JLabel** (*Etiqueta*) sobre la ventana hasta el punto indicado.



- Localice la propiedad **text** de la etiqueta, haga click a su derecha, borre el texto, escriba **Tamaño** y pulse enter.

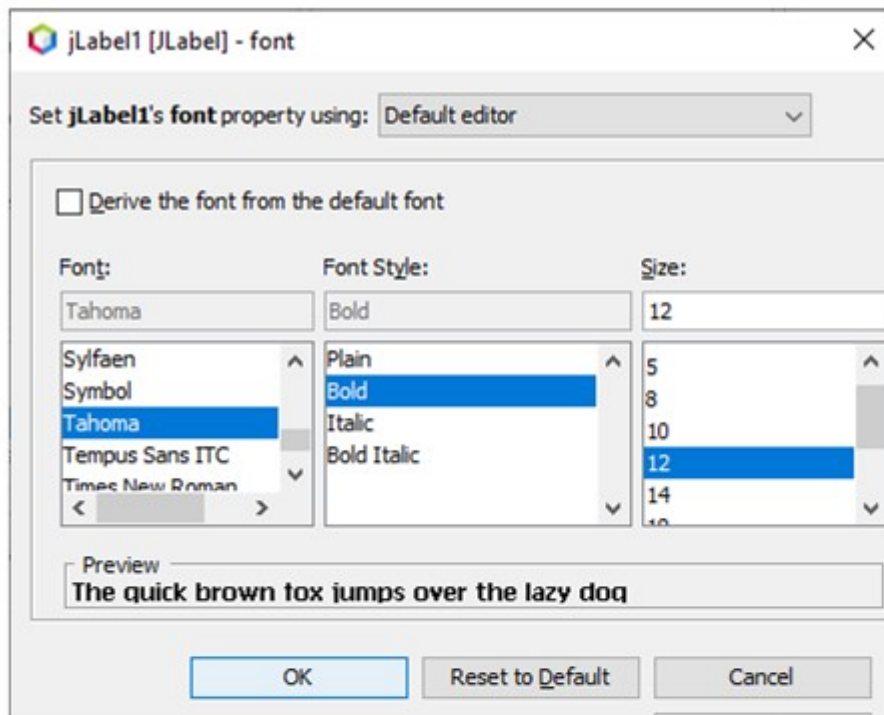


- Para poner en negrilla un texto, hacemos click en el botón de tres puntos, que se encuentra a la derecha de la propiedad **font**:

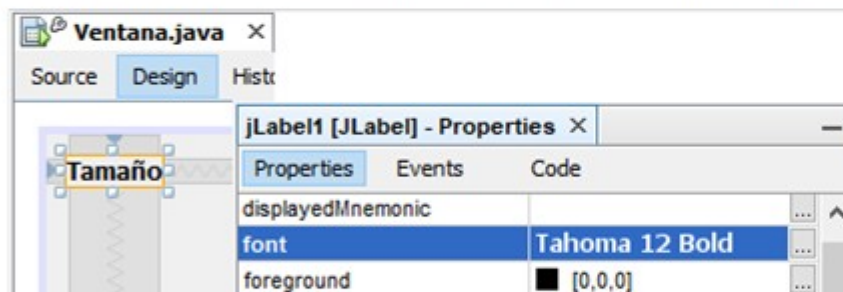


Hacer click en este boton

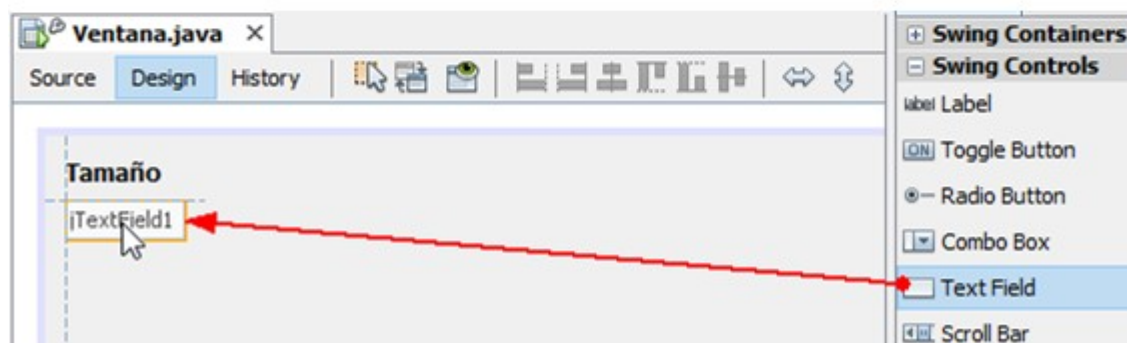
- ➡ Con ello se despliega el siguiente cuadro de dialogo, en el que puede escoger el tipo de letra, tamaño y estilos. Una vez hecho esto haga click en el botón **Aceptar**.



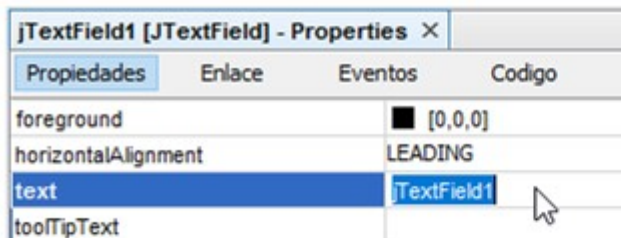
- ➡ Con ello observara el cambio en la propiedad, que ya muestra el estilo *Negrita*:



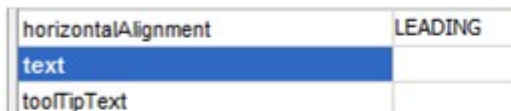
- ➡ Arrastre el campo de texto (**JTextField**) hasta la ventana en la posición indicada:



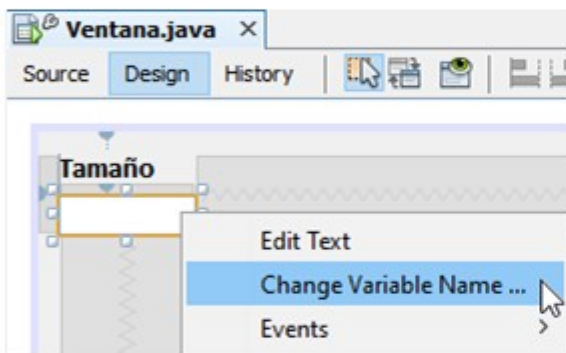
- Estando seleccionado el campo de texto, busque la propiedad **text** en el inspector de propiedades; haga click a su derecha para activar la edición del texto



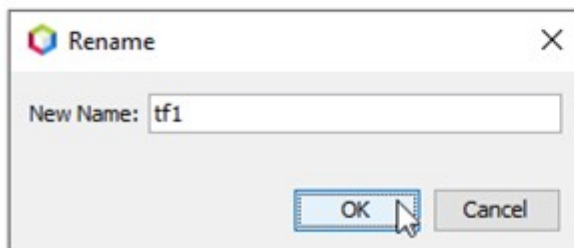
- Seguidamente borre el texto y pulse la tecla enter:



- Si es necesario, redimensione el campo de texto, luego haga click derecho sobre él y seleccione la opción "Cambiar nombre de variable ..."

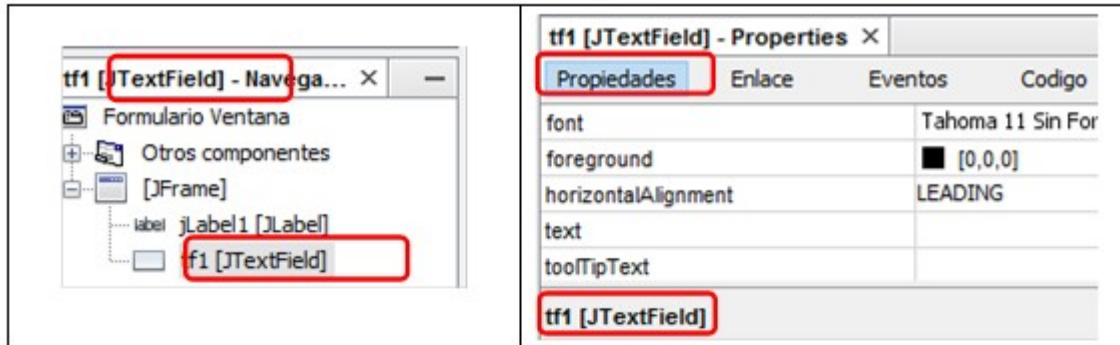


- Así se muestra un cuadro de dialogo, en el que ponemos el nombre **tf1** para el control y después haga click en el botón **OK**.

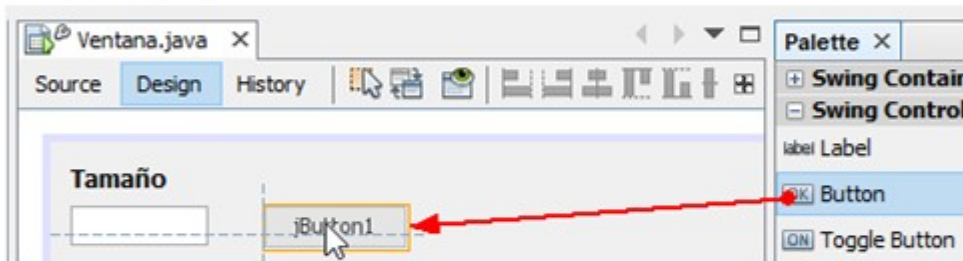


- Recuerde que el cambio de esta propiedad no afecta ni el comportamiento ni el aspecto del control, y solo es usado para acceder de forma precisa al control en la programación de los eventos.

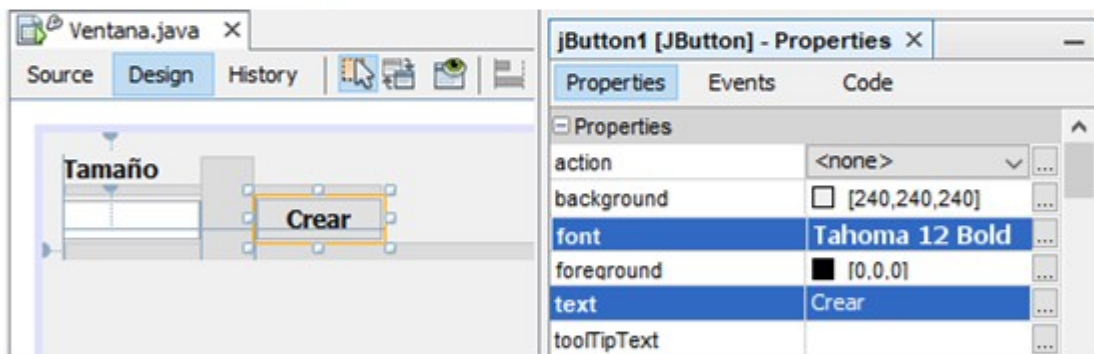
- ➡ Hecho esto evidenciamos el cambio, tanto en el navegador de componentes (imagen de la izquierda) como en el inspector de propiedades (imagen de la derecha), que encerramos en rojo aquí:



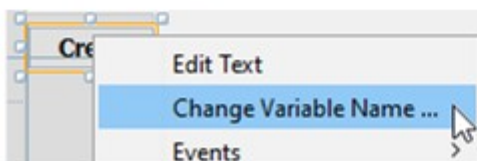
- ➡ Arrastre un botón (**JButton**) hasta la ventana en el punto indicado:



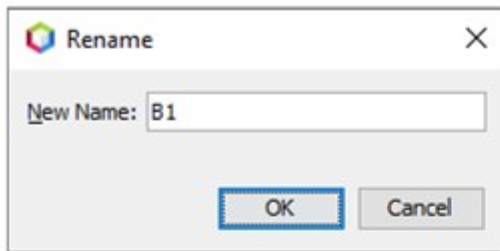
- ➡ Luego cambie la propiedad **text** del botón asignándole por valor el texto **Crear** y ponga el texto en negrilla:



- ➡ Seguidamente haga click derecho sobre el botón y seleccione la opción "Cambiar nombre de variable ...".



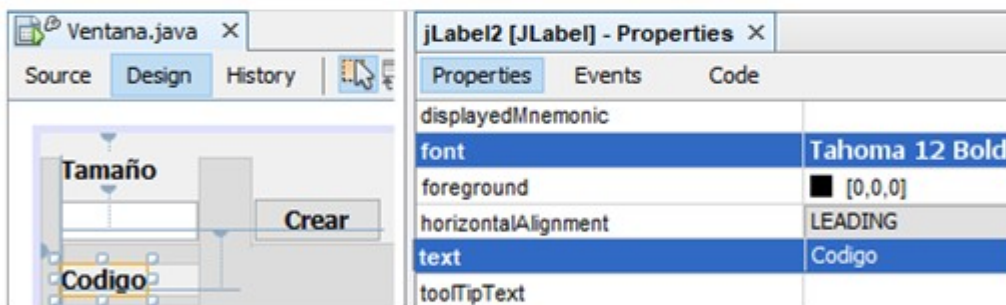
- ➡ En el cuadro de dialogo mostrado ponga **B1** al nombre del botón.



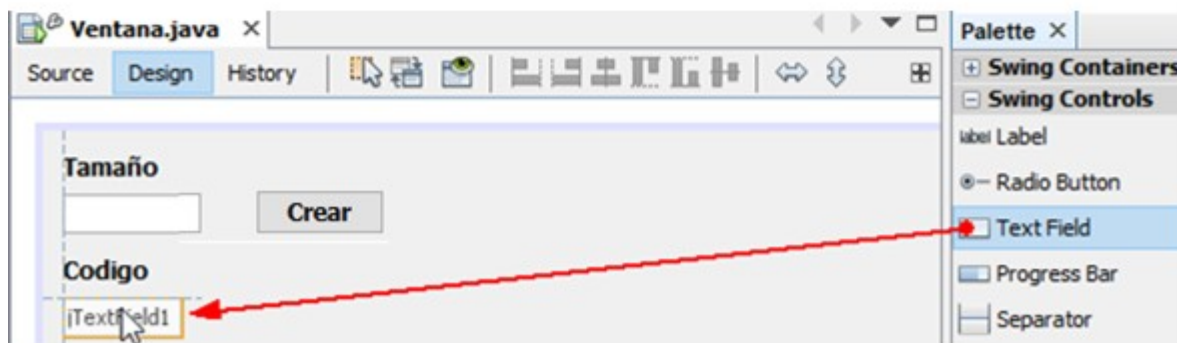
- ➡ Luego arrastre hasta la ventana otra etiqueta (**JLabel**) y déjelo en la posición indicada.



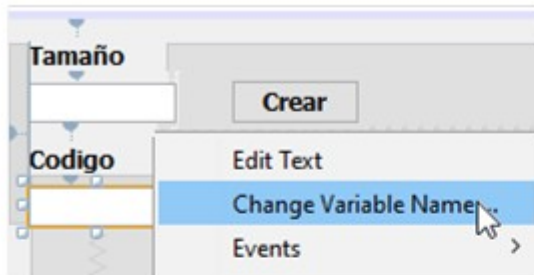
- ➡ Modifique la propiedad **text** de la etiqueta, poniéndole **Código** y ponga el texto en negrilla:



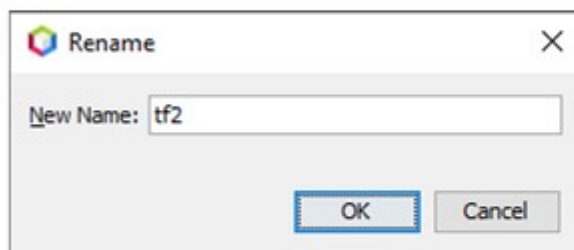
- ➡ Arrastre el campo de texto (**JTextField**) hasta la ventana en la posición indicada:



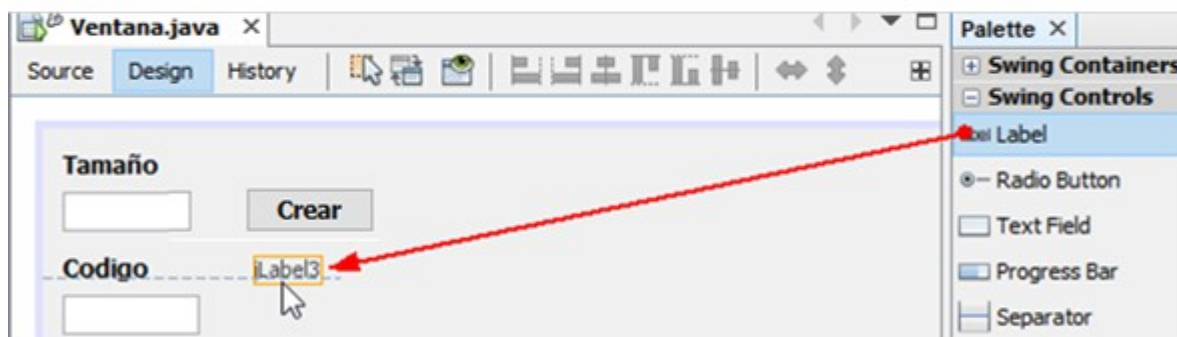
- Seguidamente borre la propiedad **text** del control, pulse la tecla enter, rediménsionelo y luego haga click derecho sobre él y seleccione la opción "Cambiar nombre de variable ...".



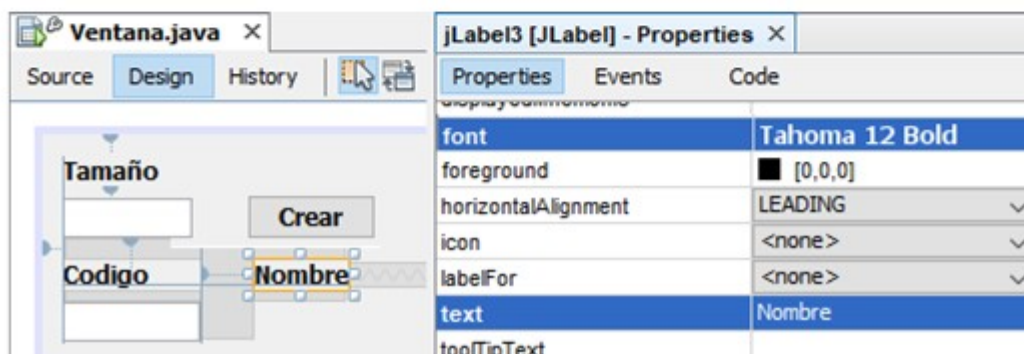
- Ahora ponga el nombre **tf2** para el control y después haga click en el botón **OK**.



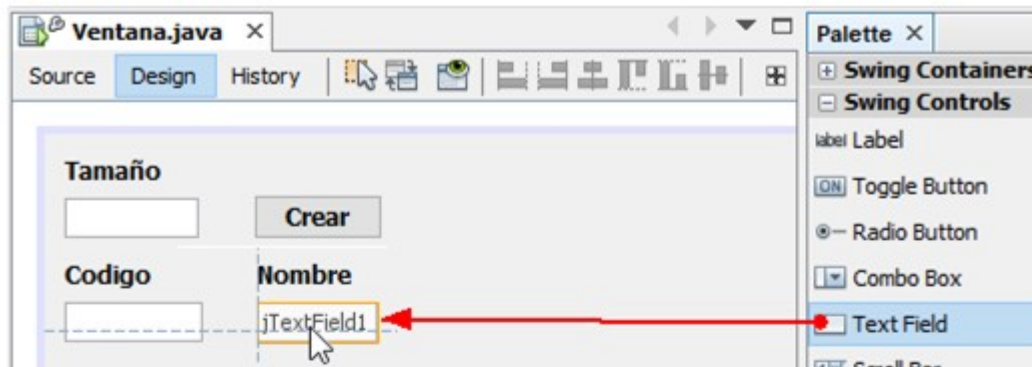
- Después arrastre otra etiqueta y déjela en la posición indicada.



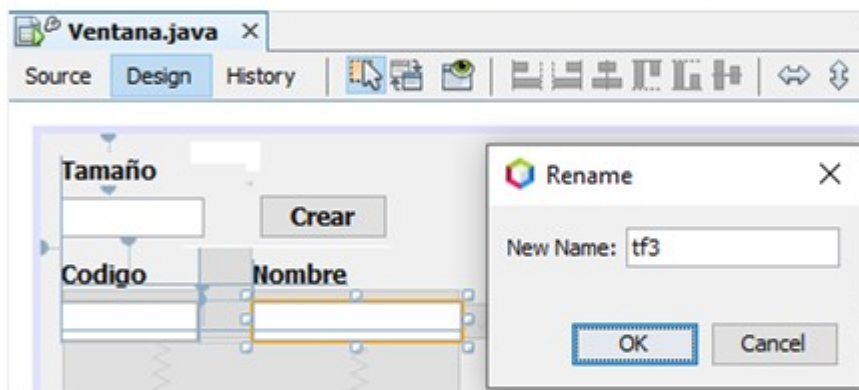
- Asigne **Nombre** a la propiedad **text** de la etiqueta y póngala en negrilla al texto:



- Arrastre el campo de texto (***JTextField***) hasta la ventana en la posición indicada:



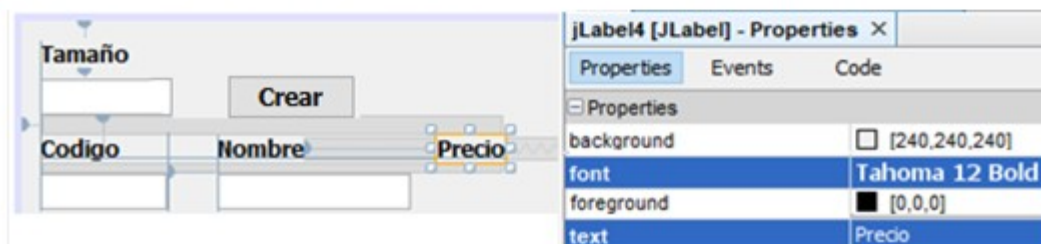
- Seguidamente borre la propiedad ***text*** del control, pulse la tecla enter, redimENSIONÉLO y luego haga click derecho sobre él y seleccione la opción "*Cambiar nombre de variable ...*" y asigne ***tf3*** por nombre.



- Arrastre otra etiqueta (***JLabel***) y déjela en la posición indicada.



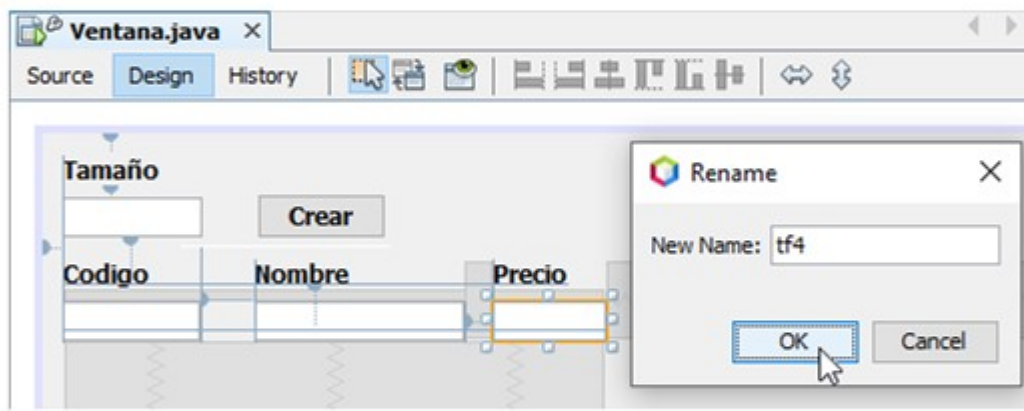
- En la propiedad ***text*** asígnele ***Precio*** y ponga en negrilla el texto:



- ➡ Arrastre otro campo de texto hasta la ventana en la posición indicada:



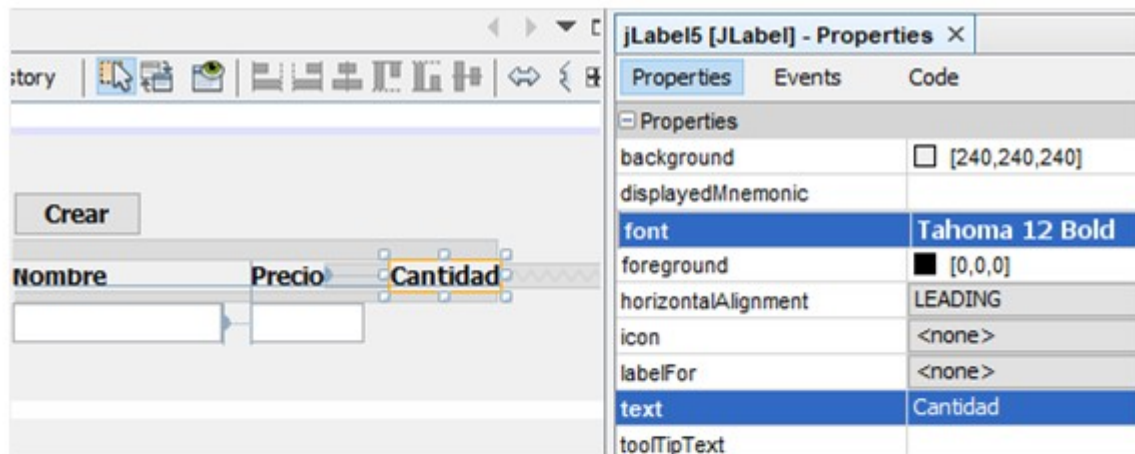
- ➡ Luego borre la propiedad **text** del control, rediménsionelo y luego haga click derecho sobre él, tome la opción "Cambiar nombre de variable ..." y asigne **tf4** por nombre.



- ➡ Arrastre otra etiqueta (**JLabel**) y déjela en la posición señalada.



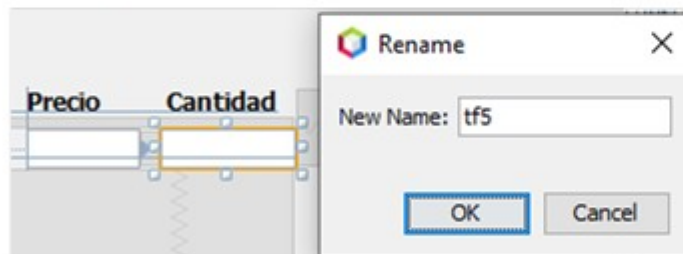
- ➡ Ahora en la propiedad **text** de la etiqueta asigne **Cantidad** y ponga en negrilla el texto:



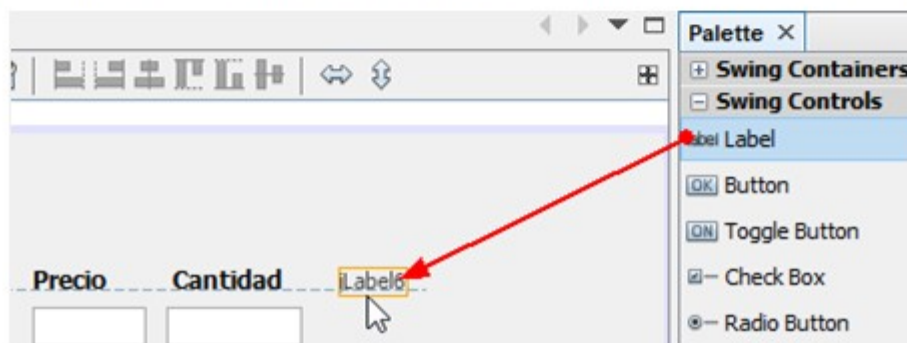
- ➡ Arrastre otro campo de texto hasta la ventana en la posición indicada:



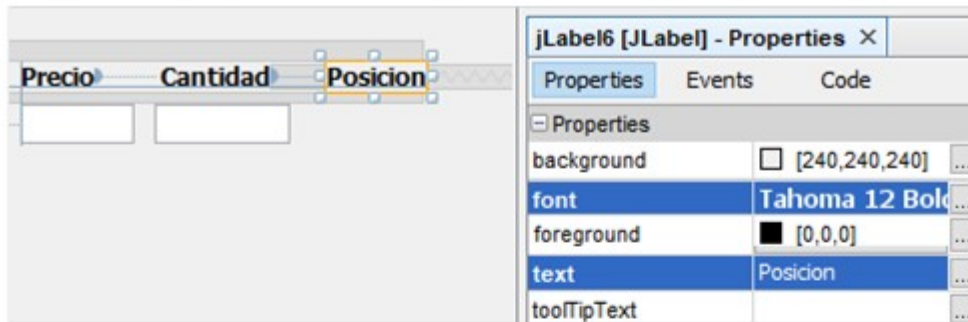
- ➡ Luego borre la propiedad **text** del control, redimENSIONÉLO y luego haga click derecho sobre él, tome la opción "Cambiar nombre de variable ..." y asigne **tf5** como nombre para el control.



- ➡ Después arrastre una etiqueta hasta el punto indicado de la ventana:



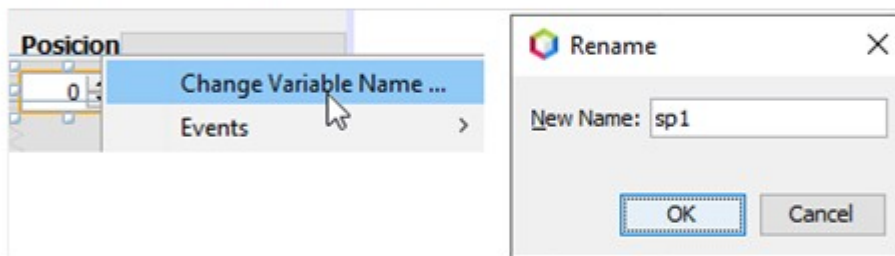
- ➡ Luego en la propiedad **text** asigne **Posición** y ponga en negrilla el texto:



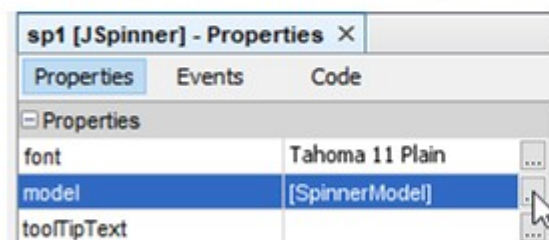
- ➡ Después arrastre hasta este punto un control **JSpinner**.



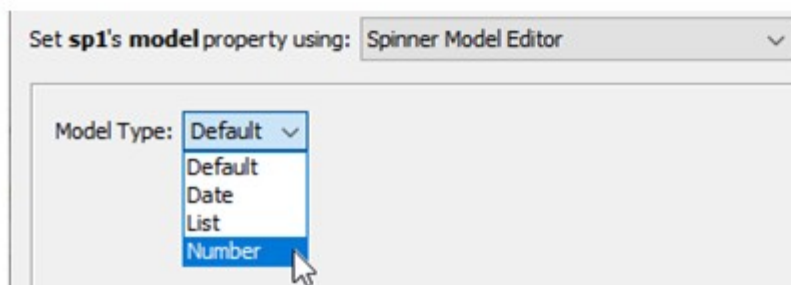
- ➡ Haga click derecho sobre este control y tome la opción "*Cambiar nombre de variable ...*" y asigne **sp1** como nombre.



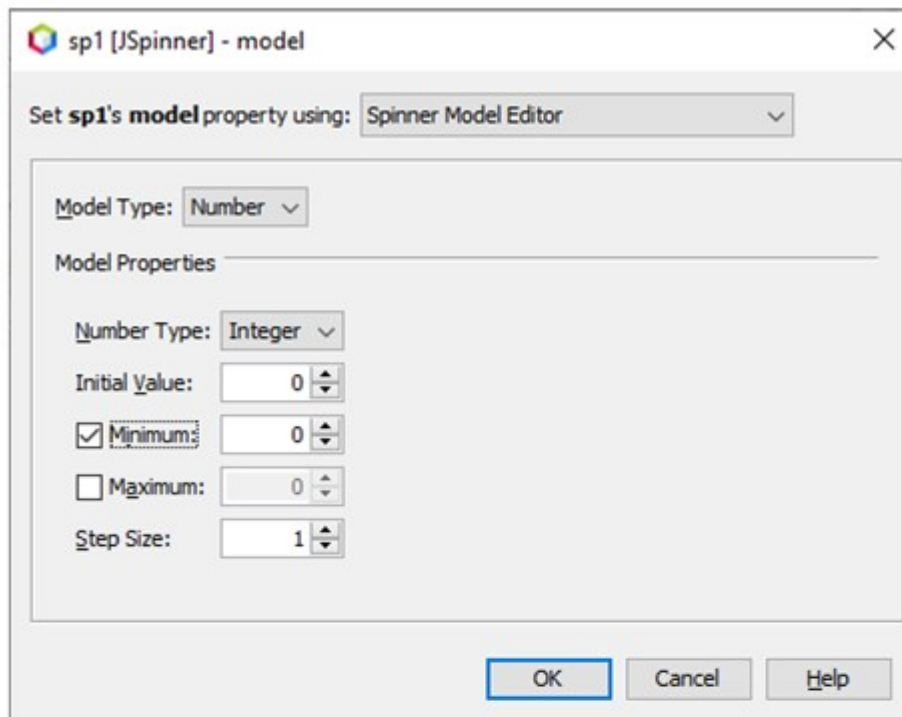
- ➡ En la propiedad **model** del **JSpinner** haga click en el botón de tres puntos (...).



- ➡ En el tipo de modelo escoja **Number**.



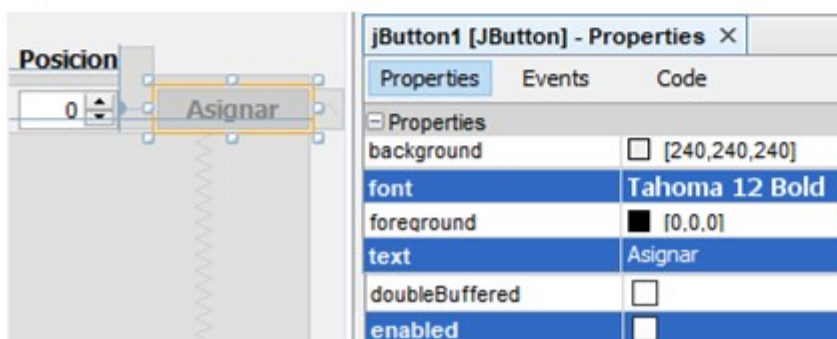
- Luego en **Number Type** seleccione **Integer**, marque la casilla **Minimum** y póngale el valor de 0; finalmente haga click en el botón **OK**



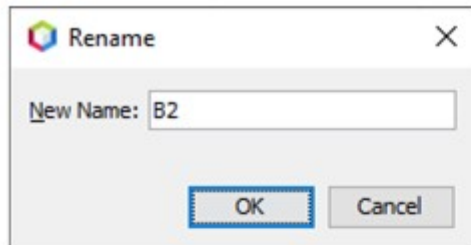
- Ahora arrastre un botón (**JButton**) hasta la ventana en el punto indicado:



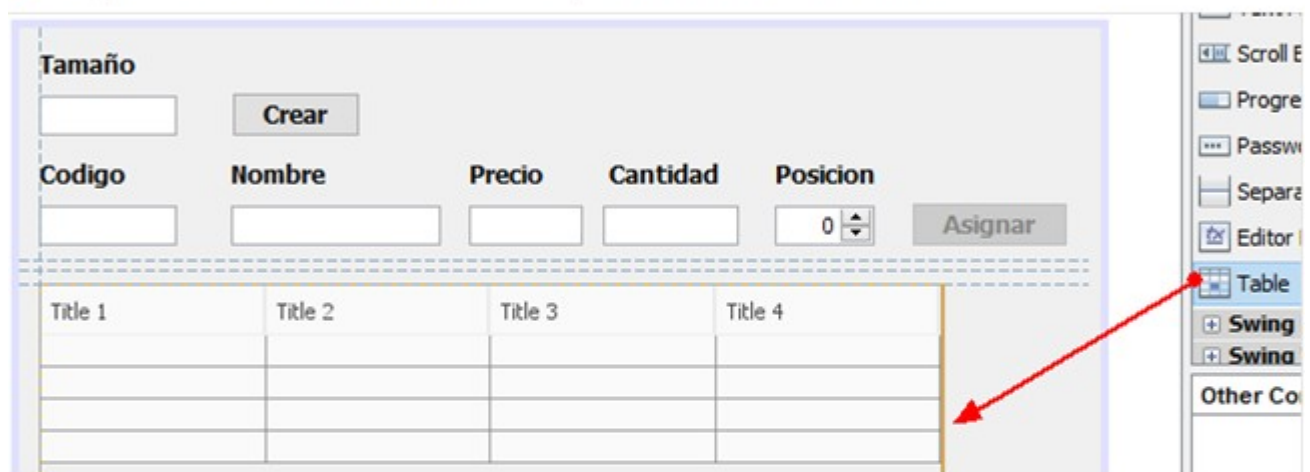
- Cambie la propiedad **text** del botón póngale **Asignar**, asigne negrilla al texto y desmarque la casilla que está a la derecha de la propiedad **enabled**, lo cual lo deja inhabilitado.



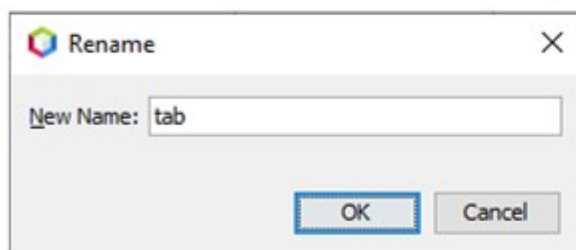
- Seguidamente haga click derecho sobre el botón y seleccione la opción "*Cambiar nombre de variable ...*" y en el cuadro de dialogo mostrado ponga **B2** al nombre del botón.



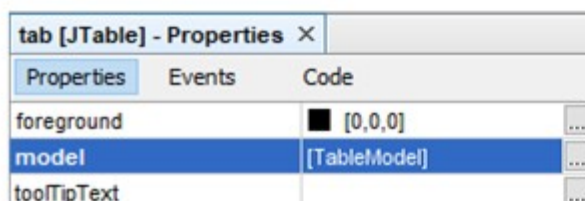
- Después arrastre un **JTable** hasta la parte inferior de la ventana:



- Ahora haga click derecho sobre la tabla, seleccione la opción "*Cambiar nombre de variable ...*" y en el cuadro de dialogo mostrado póngale por nombre **tab**.



- Luego en la propiedad **model** del **JTable** haga click en el botón de tres puntos (...) que se encuentra a la derecha.



- ➔ En la ventana desplegada ponga 0 como numero de filas (**Rows**), 4 como numero de columnas (**Columns**); defina cada columna con los títulos y tipos indicados en la siguiente imagen, además desmarque todas las casillas de la columna **Editable** para que las celdas de la tabla sean de solo lectura.

Set **tab's model** property using: Table model customizer

Table Model

Table Settings Default Values

Specify Title and Column Types Here:

Column	Title	Type	Editable
1	Codigo	Integer	<input type="checkbox"/>
2	Nombre	String	<input type="checkbox"/>
3	Precio	Integer	<input type="checkbox"/>
4	Cantidad	Integer	<input type="checkbox"/>

Rows: + - Columns: + -

Insert Delete Move Up Move Down

OK Reset to Default Cancel Help

- ➔ Finalmente haga click en el botón **OK**, quedando la tabla así:

Ventana.java

Source Design History

Tamaño

Crear

Codigo **Nombre** **Precio** **Cantidad** **Posicion**

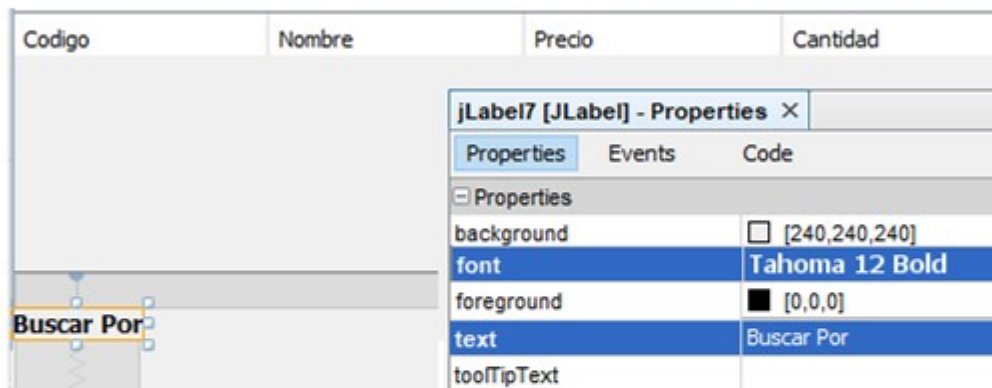
Asignar

Codigo	Nombre	Precio	Cantidad

- Seguidamente arrastre un **JLabel** hasta la ventana en el punto indicado:



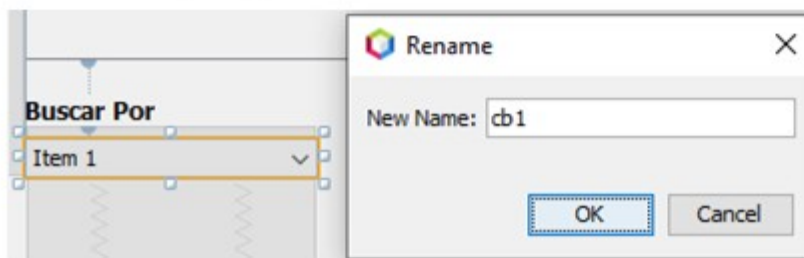
- Luego en la propiedad **text** asigne **Buscar Por** y ponga en negrilla el texto:



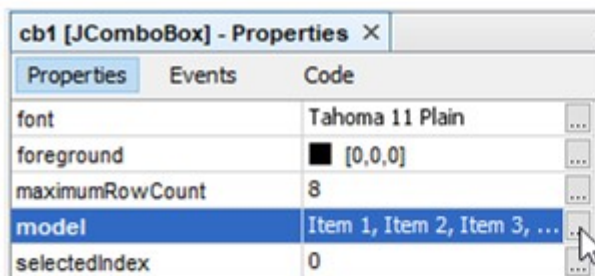
- Después arrastre hasta este punto un control **JComboBox** (Lista Desplegable).



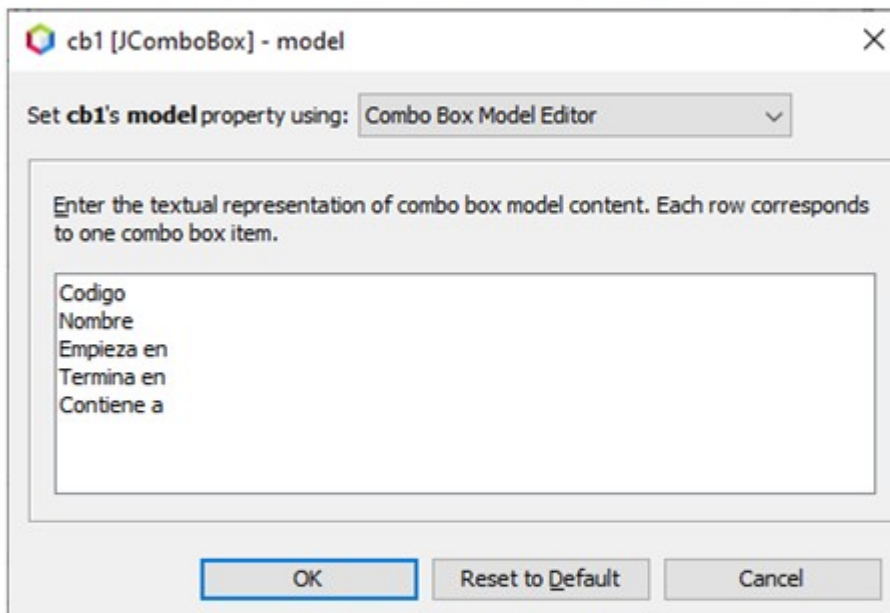
- Haga click derecho sobre este control y tome la opción "Cambiar nombre de variable ..." y asigne **cb1** como nombre.



- ➡ En la propiedad **model** del **JComboBox** haga click en el botón de tres puntos (...) que se encuentra a la derecha.



- ➡ Luego borre todos los ítems que aparecen en el cuadro de dialogo desplegado, en cambio escriba las opciones mostradas en la imagen de abajo:



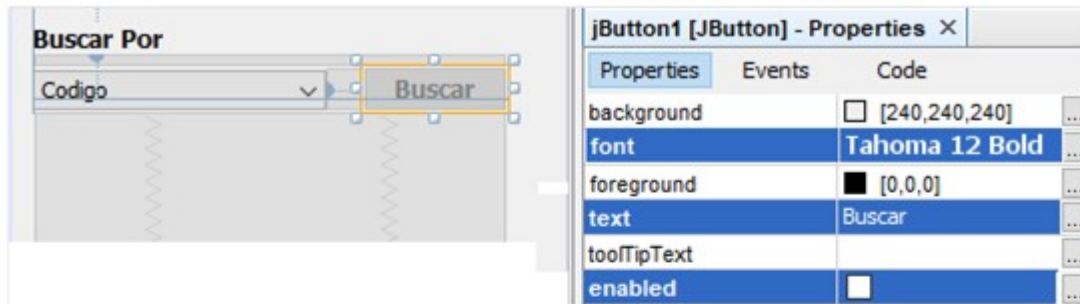
- ➡ Haga click en el botón OK y el control queda con el siguiente aspecto:



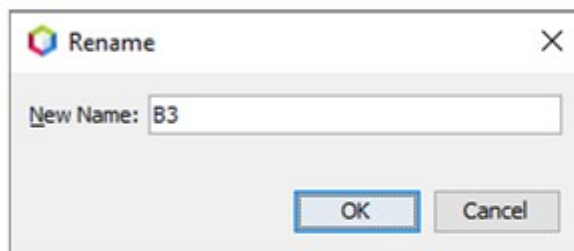
- ➡ Ahora arrastre un **JButton** hasta el punto indicado:



- Cambie la propiedad **text** del botón póngale **Buscar**, asigne negrilla al texto y desmarque la casilla que está a la derecha de la propiedad **enabled**, para inhabilitarlo.



- Seguidamente haga click derecho sobre el botón, seleccione la opción "Cambiar nombre de variable ..." y en el cuadro de dialogo mostrado ponga **B3** al nombre del botón.



8. Implementación atributos y metodos de la ventana.

- Vaya al código fuente de la ventana, haciendo click en la pestaña **Source** (**Fuente**)



- Ubique la línea de declaración de la clase (`public class Ventana extends javax.swing.JFrame {`) y antes de ella importe las tres siguientes clases (solo el código del marco rojo):

```
import javax.swing.JOptionPane;
import javax.swing.SpinnerNumberModel;
import javax.swing.table.DefaultTableModel;

public class Ventana extends javax.swing.JFrame {
```

- ➡ Luego debajo de la línea de declaración de la clase (`public class Ventana extends javax.swing.JFrame {`) añada las siguientes instancias como atributos de la ventana (solo el código del marco rojo):

```
public class Ventana extends javax.swing.JFrame {  
  
    private TVecProducto VP;  
    private DefaultTableModel TaMod;  
    private SpinnerNumberModel SpMod;  
  
}
```

- ➡ Un poco más abajo ubique al método constructor de la ventana, que en este punto debe estar así:

```
public Ventana() {  
    initComponents();  
}
```

- ➡ Modifique la implementación de este constructor, agregando (debajo del llamado al método *initComponents*) las inicializaciones de las instancias declaradas antes (solo el código del marco rojo):

```
public Ventana() {  
    initComponents();  
  
    VP=new TVecProducto();  
    TaMod=(DefaultTableModel) tab.getModel();  
    SpMod=(SpinnerNumberModel) spl.getModel();  
  
}
```

- ➡ Debajo de la implementación de este método (después de la llave de cierre `}` de este método constructor *Ventana*), agregue la codificación de los siguientes métodos:

```
private void Limpiar() {  
    tf2.setText("");  
    tf3.setText("");  
    tf4.setText("");  
    tf5.setText("");  
    tf2.grabFocus();  
}
```



```

private void VerMensaje(String Texto) {
    JOptionPane.showMessageDialog(null, Texto);
}

private void Llenar(TProducto Pro) {
    Pro.setCodigo(Integer.parseInt(tf2.getText()));
    Pro.setNombre(tf3.getText());
    Pro.setPrecio(Integer.parseInt(tf4.getText()));
    Pro.setCantidad(Integer.parseInt(tf5.getText()));
}

private void Mostrar(TProducto Pro) {
    tf2.setText(Pro.getCodigo()+"");
    tf3.setText(Pro.getNombre());
    tf4.setText(Pro.getPrecio()+"");
    tf5.setText(Pro.getCantidad()+"");
}

private void Mostrar() {
    int i;
    TProducto Pro;
    for(i=0;i<VP.getTam();i++){
        Pro=VP.getVec(i);
        TaMod.setValueAt(Pro.getCodigo(),i,0);
        TaMod.setValueAt(Pro.getNombre(),i,1);
        TaMod.setValueAt(Pro.getPrecio(),i,2);
        TaMod.setValueAt(Pro.getCantidad(),i,3);
    }
}

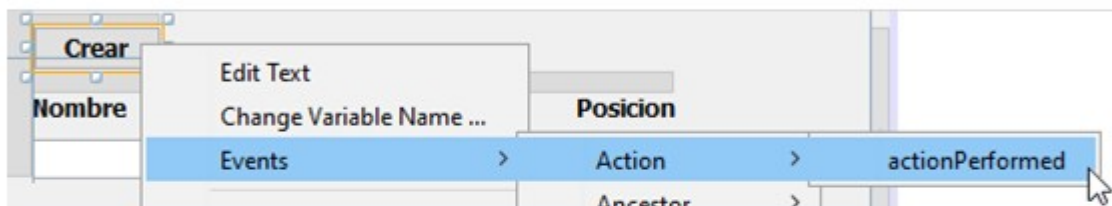
private void Mostrar(TProducto Pro,String Error) {
    if(Pro!=null){
        Mostrar(Pro);
    }
    else{
        VerMensaje(Error);
    }
}

```

```
private void Mostrar(int Pos,String Error) {
    if(Pos>=0) {
        Mostrar(VP.getVec(Pos));
    }
    else{
        VerMensaje(Error);
    }
}
```

9. Implementación eventos de la ventana.

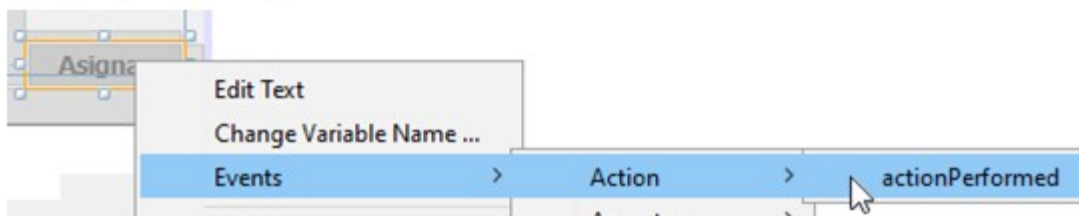
- Ahora vaya al diseño de la ventana, haga click derecho sobre el botón **Crear**, tome la opción **Eventos + Action + ActionPerformed**



- El código para el evento click del botón **Crear** es el siguiente (solo el código del marco rojo):

```
private void B1ActionPerformed(java.awt.event.ActionEvent evt) {
    int N;
    N=Integer.parseInt(tfl.getText());
    VP.setTam(N);
    TaMod.setRowCount(N);
    SpMod.setMaximum(N-1);
    B2.setEnabled(N>0);
    B3.setEnabled(N>0);
    VerMensaje("Vector creado");
}
```

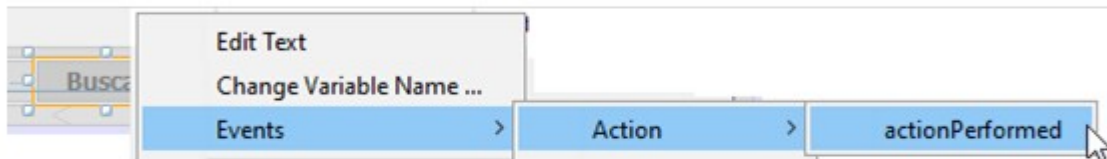
- Nuevamente vaya al diseño de la ventana, haga click derecho sobre el botón **Asignar**, tome la opción **Eventos + Action + ActionPerformed**



- ➡ El código para el evento click del botón **Asignar** es el siguiente:

```
private void B2ActionPerformed(java.awt.event.ActionEvent evt) {  
    int pos;  
    pos=(int)spl.getValue();  
    Llenar(VP.getVec(pos));  
    Mostrar();  
    Limpiar();  
    if(pos+1<VP.getTam()){  
        spl.setValue(pos+1);  
    }  
}
```

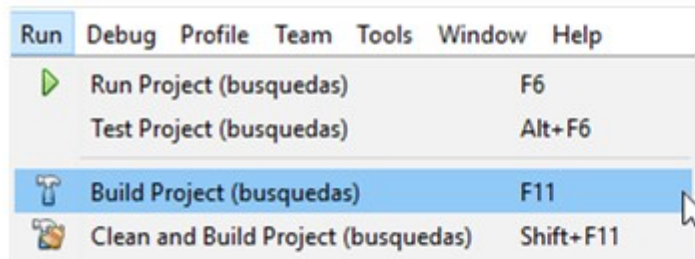
- ➡ Otra vez vaya al diseño de la ventana, haga click derecho sobre el botón **Buscar**, tome la opción **Eventos + Action + ActionPerformed**



- ➡ El código para el evento click del botón **Buscar** es el siguiente

```
private void B3ActionPerformed(java.awt.event.ActionEvent evt) {  
    int cod;  
    String nom;  
    nom=tf3.getText();  
    switch(cbl.getSelectedIndex()){  
        case 0:VP.OrdCodigo();  
            Mostrar();  
            cod=Integer.parseInt(tf2.getText());  
            Mostrar(VP.BuscarCod(cod),"Codigo " + cod + " no encontrado");  
            break;  
        case 1:Mostrar(VP.BuscarNom(nom),"Nombre (" + nom + ") no encontrado");  
            break;  
        case 2:Mostrar(VP.EmpiezaEn(nom),"Ningun nombre inicia en " + nom);  
            break;  
        case 3:Mostrar(VP.TerminaEn(nom),"Ningun nombre termina en " + nom);  
            break;  
        case 4:nom=nom + (!VP.ContieneA(nom)?" no":" si");  
            VerMensaje(nom + " esta contenido en un nombre");  
            break;  
    }  
}
```

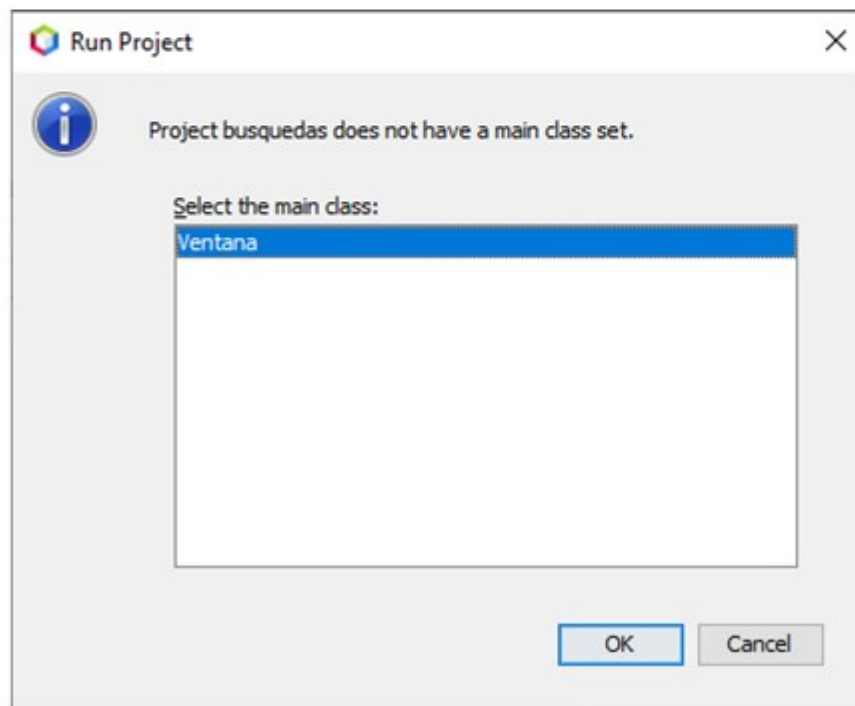
- Una vez hecho esto puede compilar el programa con la opción **Run** y luego **Build Project (búsquedas)** del menú principal o pulsando la tecla F11.



- Después puede ejecutar el programa, con la opción **Run** y **Run Project (búsquedas)** del menú principal o pulsando la tecla F6.



- Cuando se ejecuta el programa por primera vez (y solo por la primera vez), *NetBeans* le mostrará el siguiente cuadro de dialogo, en el cual sugiere usar como clase principal la clase de la ventana (el **JFrame**); por lo tanto, hacemos click en el botón **OK**, para que nuestra ventana sea la clase principal del programa:



- ➡ Las siguientes son capturas de la aplicación, en vista de diseño y de ejecución:

Tamaño

Crear

Codigo **Nombre** **Precio** **Cantidad** **Posicion**

Asignar

Codigo	Nombre	Precio	Cantidad
--------	--------	--------	----------

Buscar Por

Buscar

Ejemplo métodos de búsqueda

Tamaño

10 **Crear**

Codigo **Nombre** **Precio** **Cantidad** **Posicion**

Asignar

Codigo	Nombre	Precio	Cantidad

Buscar Por

Buscar