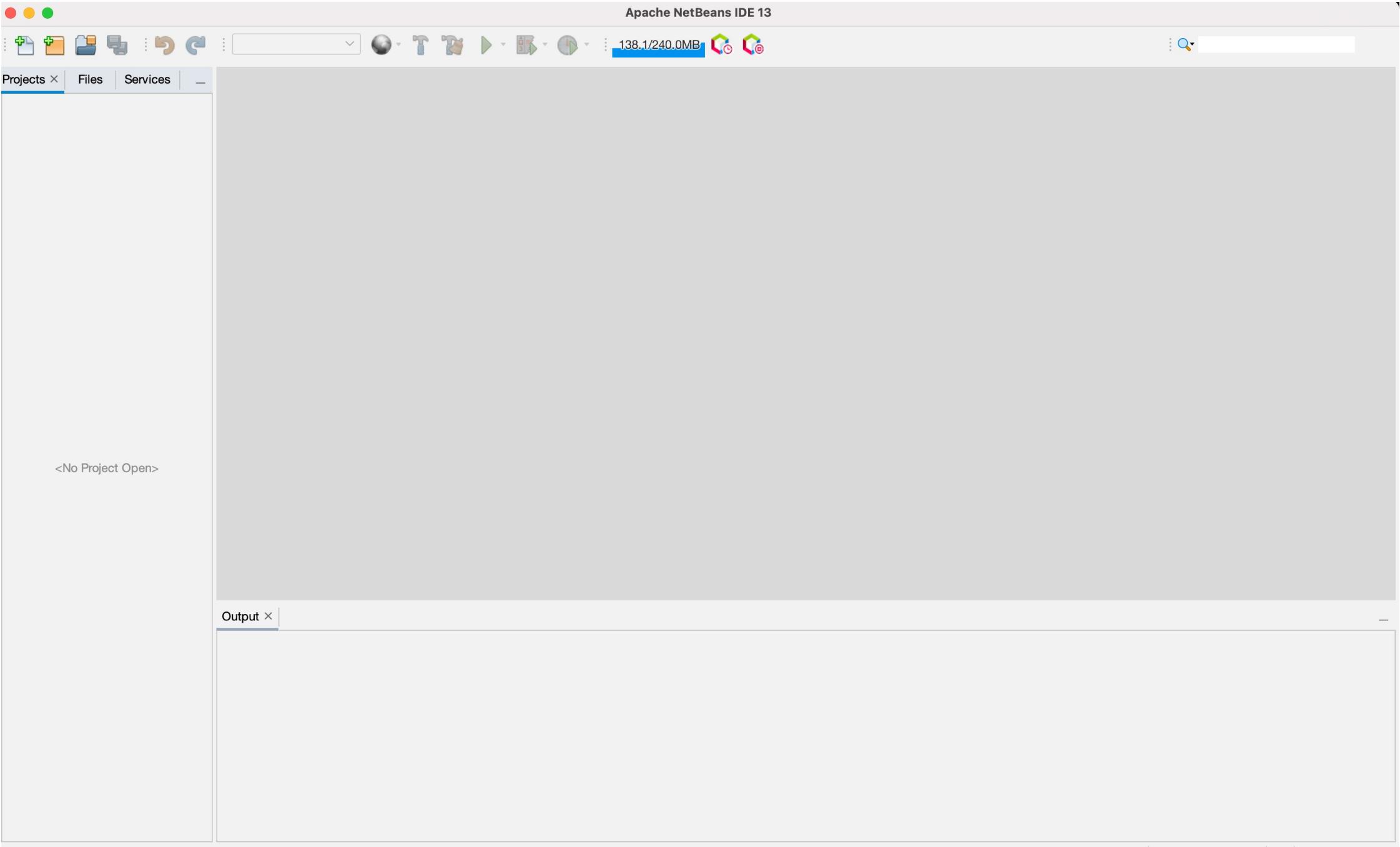
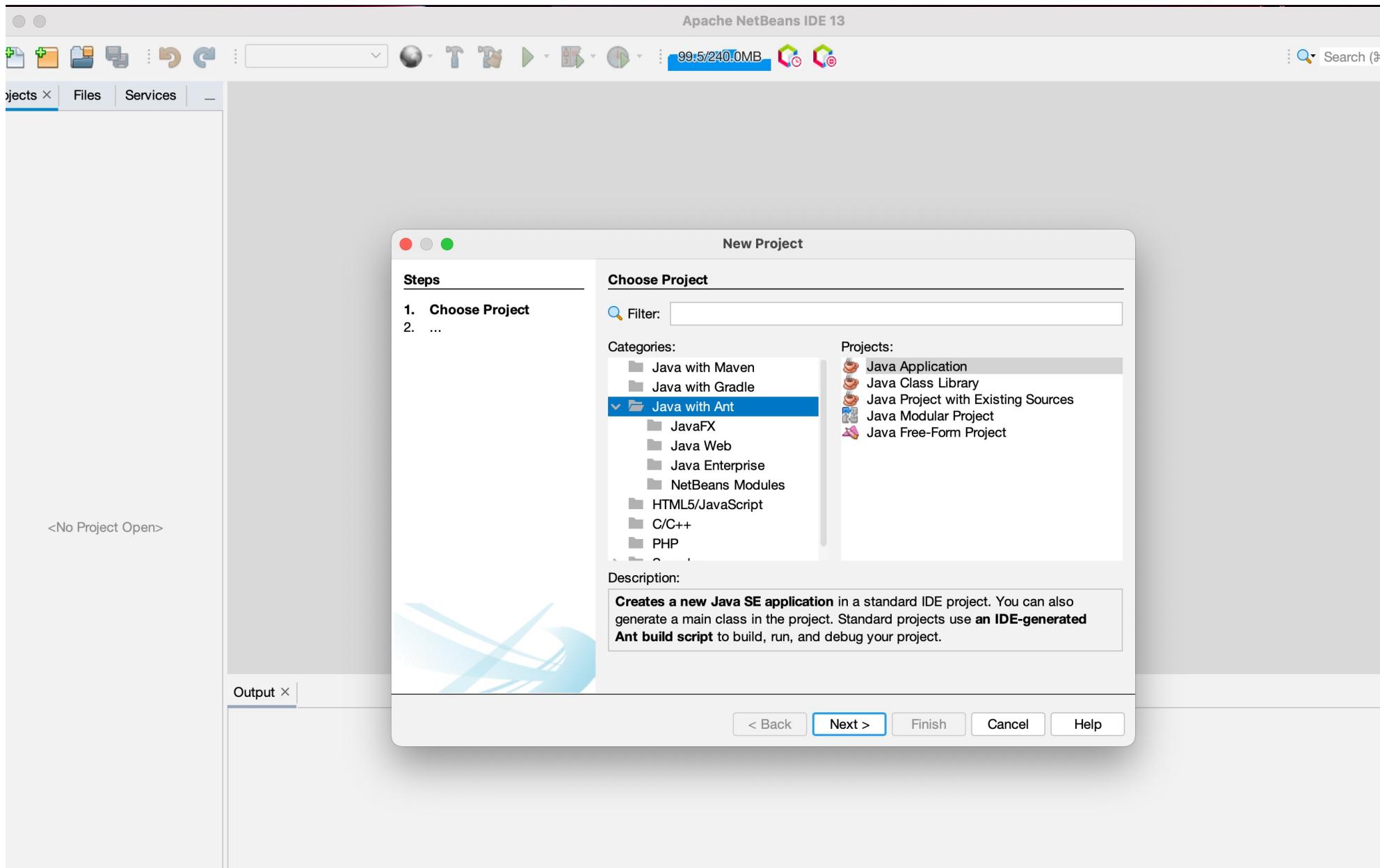


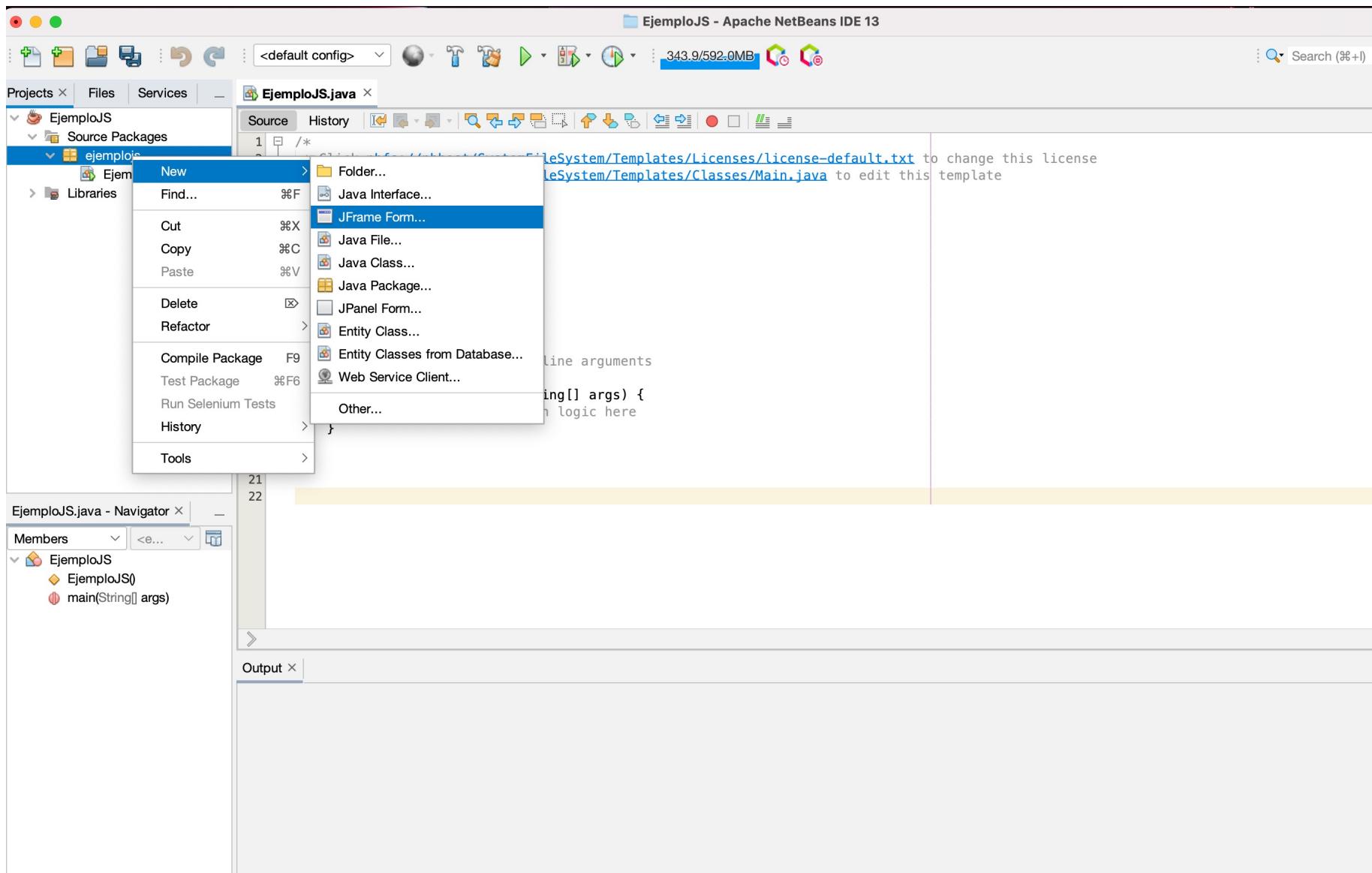
Guía práctica con NetBeans

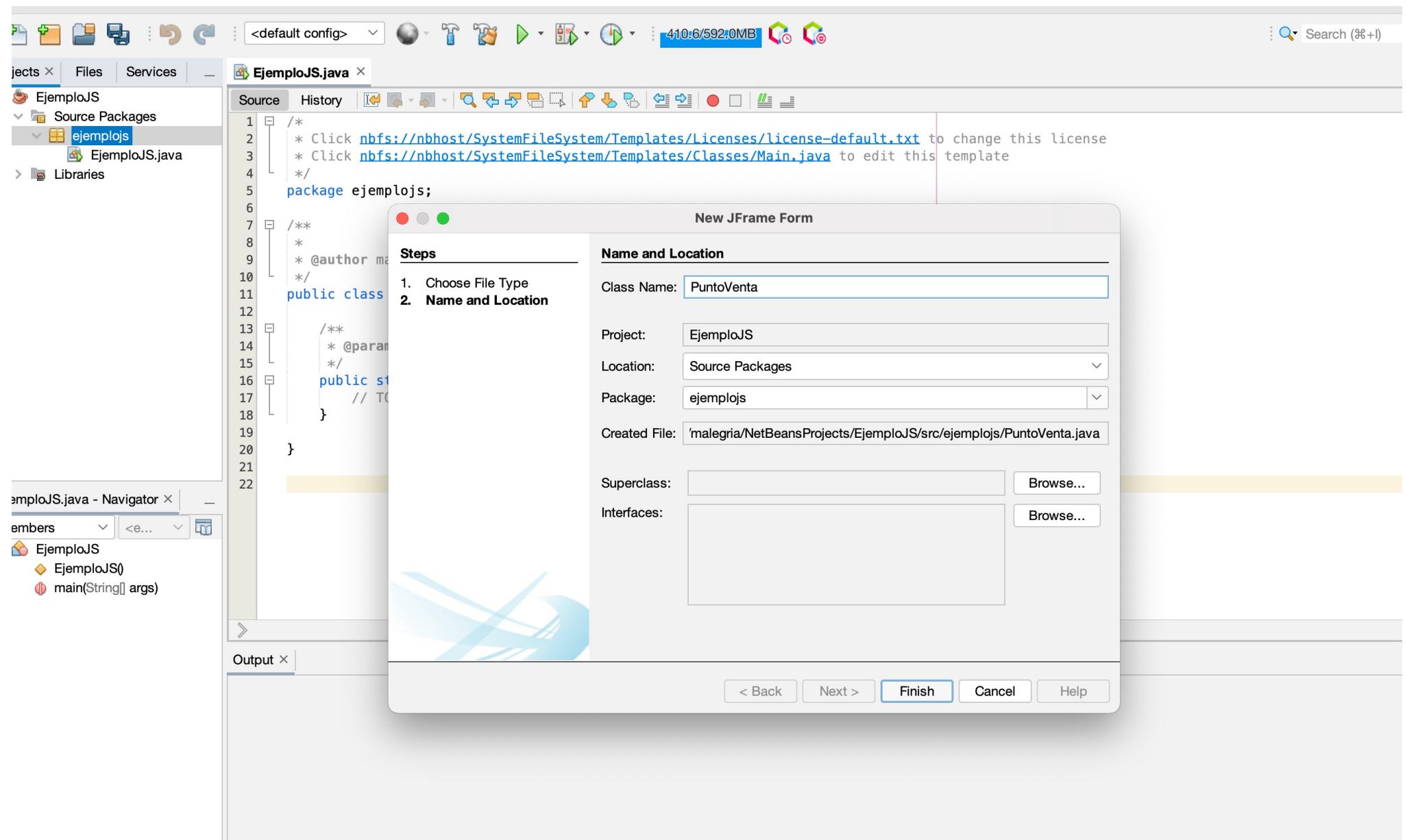


New Project → Java with Ant → Java Application

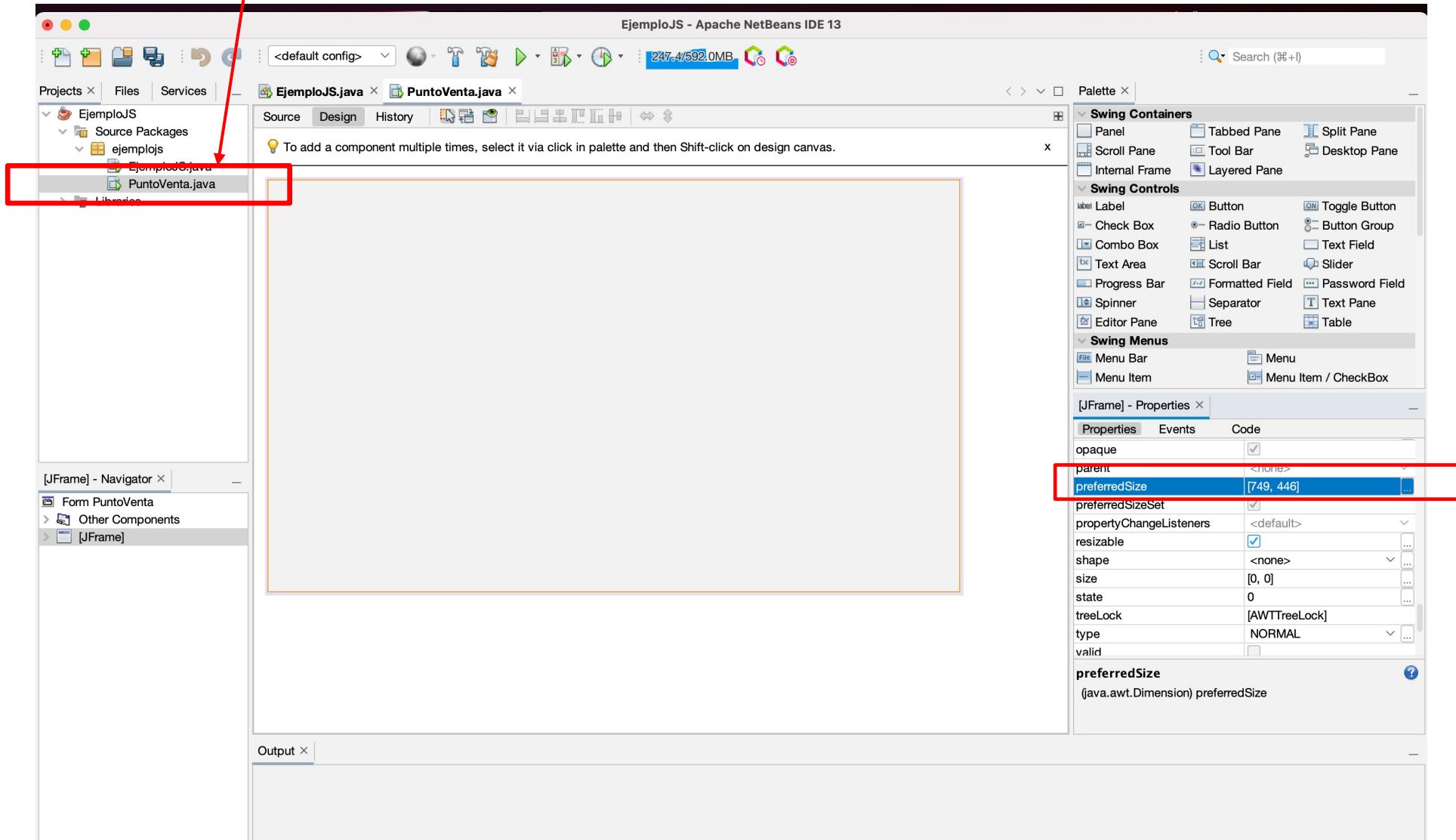


Ejemplojs → New → Jframe Form



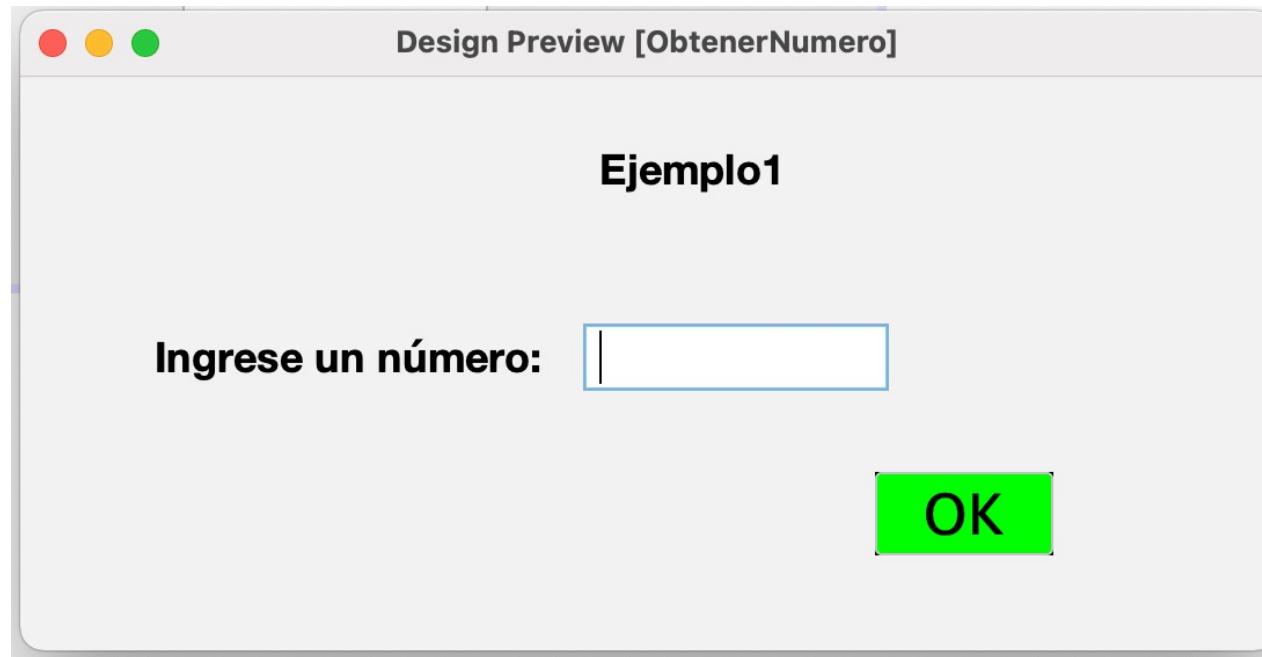


Aparece



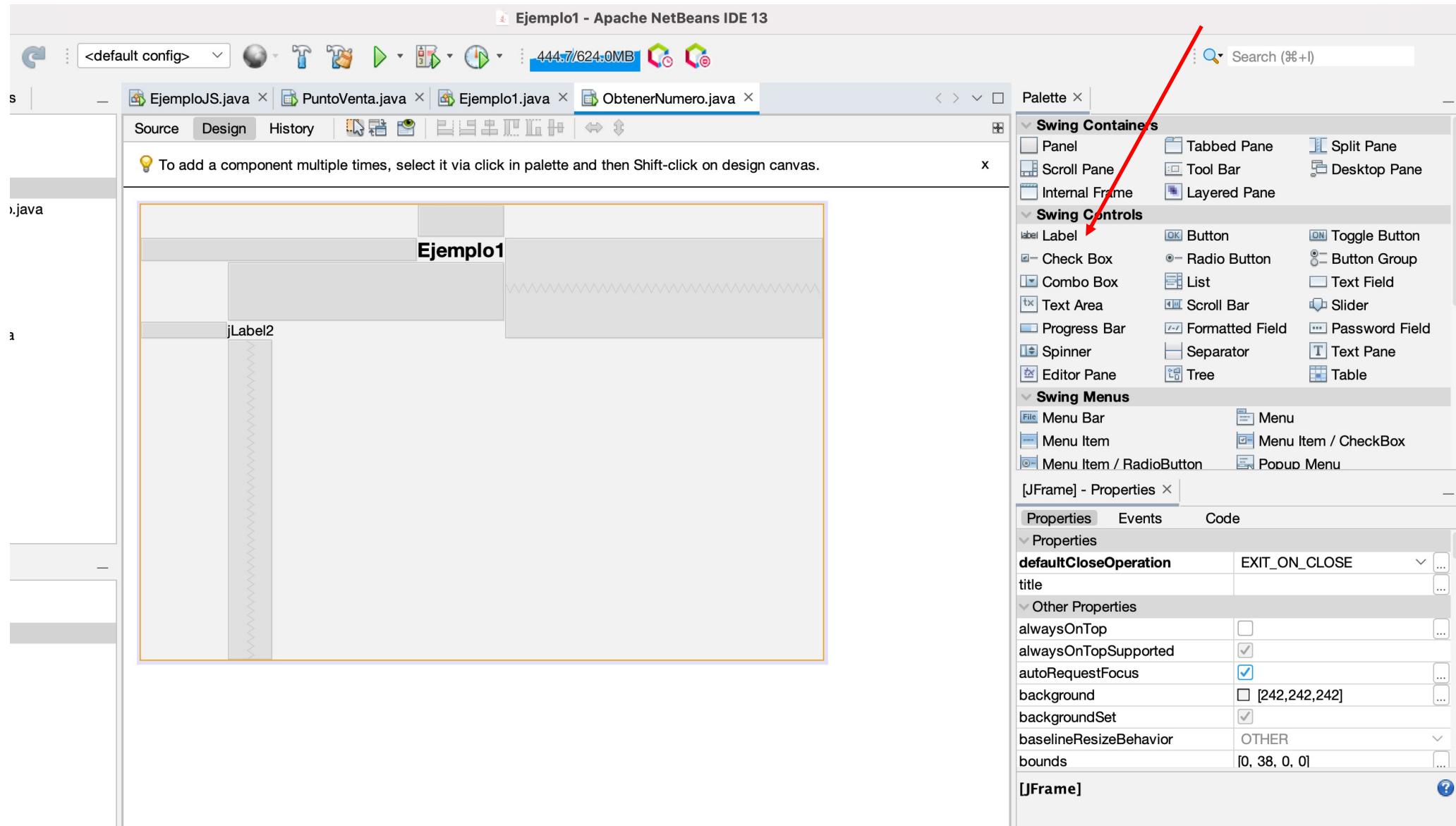
Ejemplo1:

- Pedir un valor numérico, obtenerlo y mostrarlo cuando se presiona un botón.

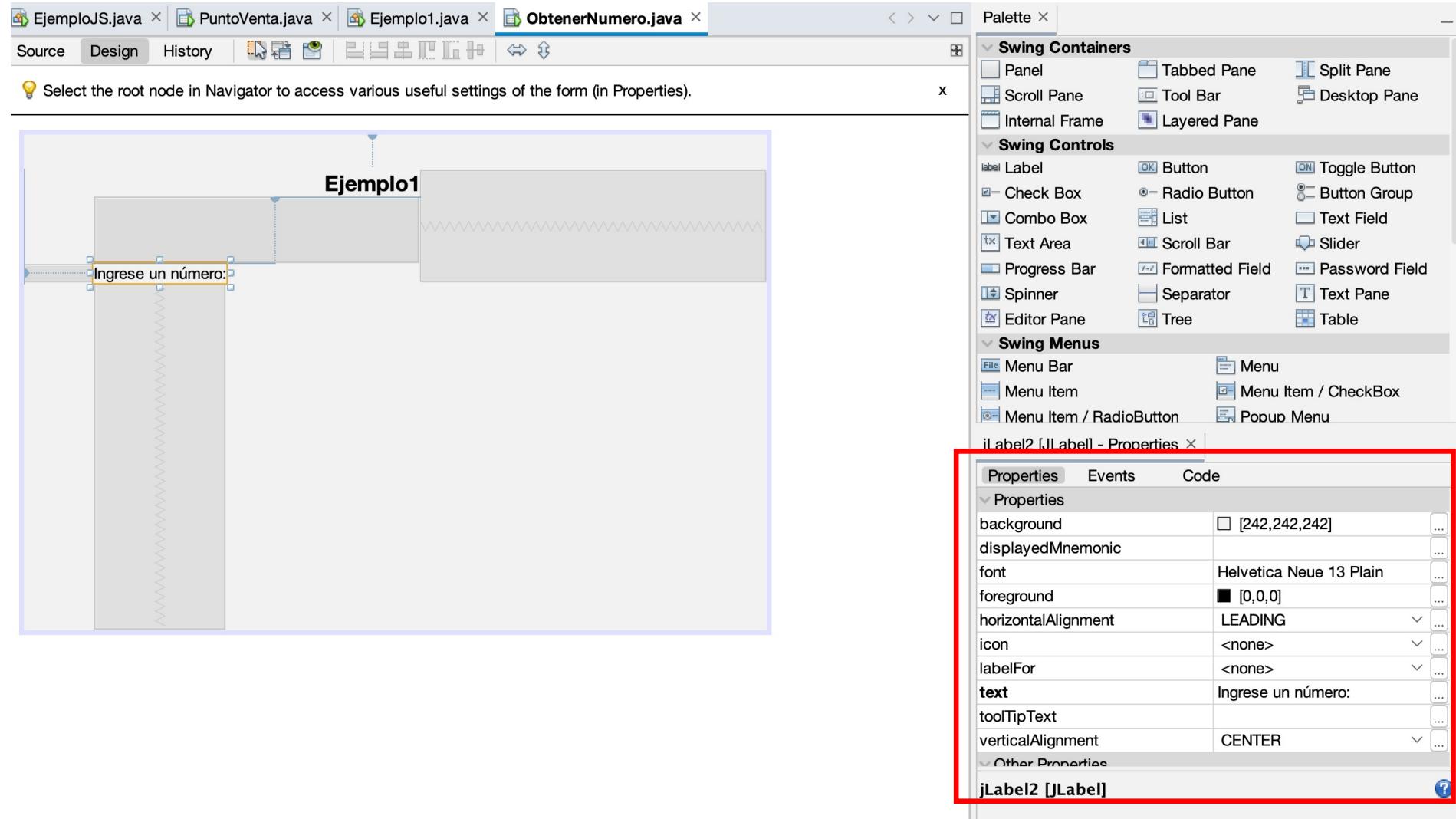


Puedes comenzar ajustando el tamaño de la ventana.

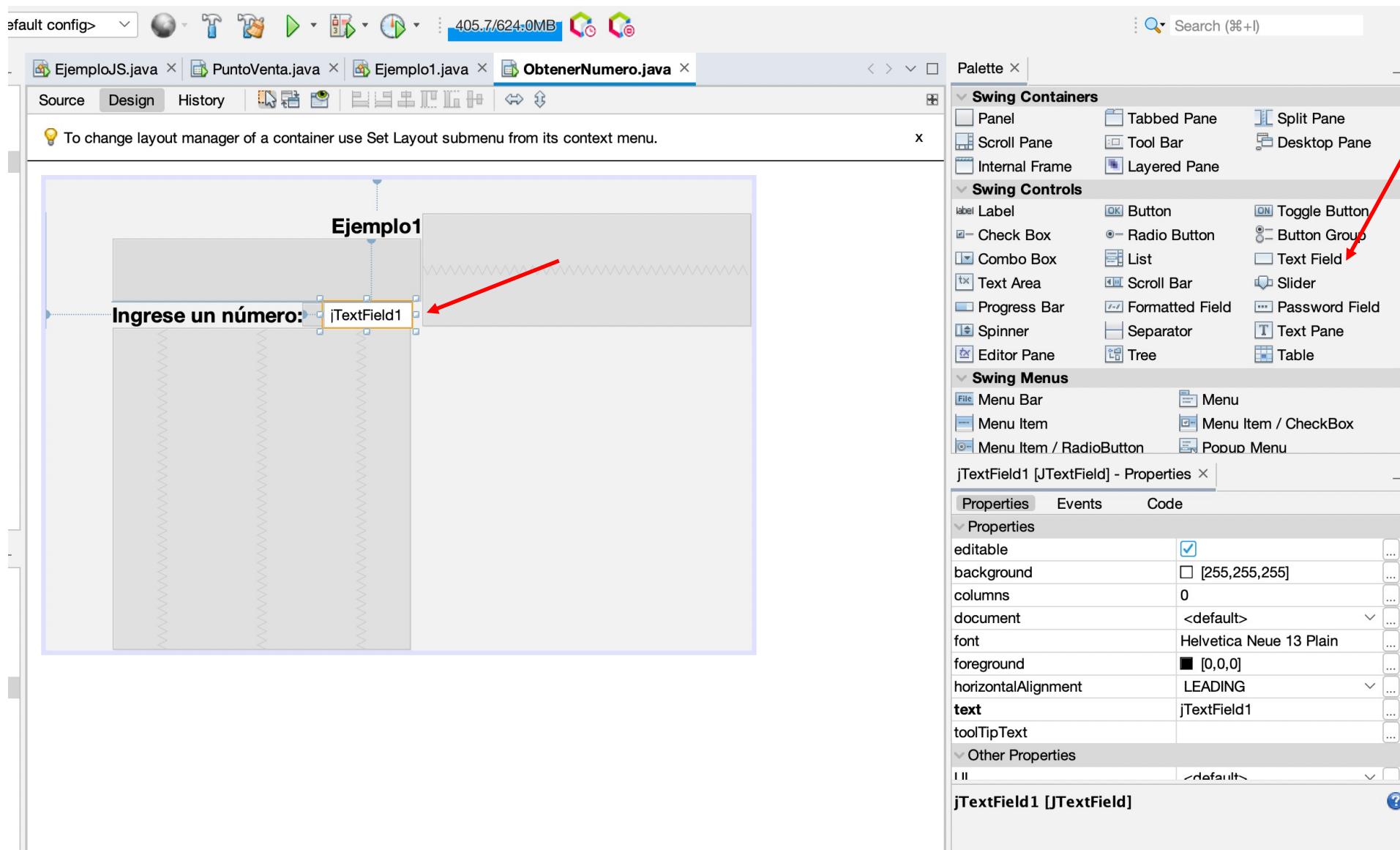
Luego puedes usar Label (Jlabel) para pedir el numero o para colocar un label en la ventana



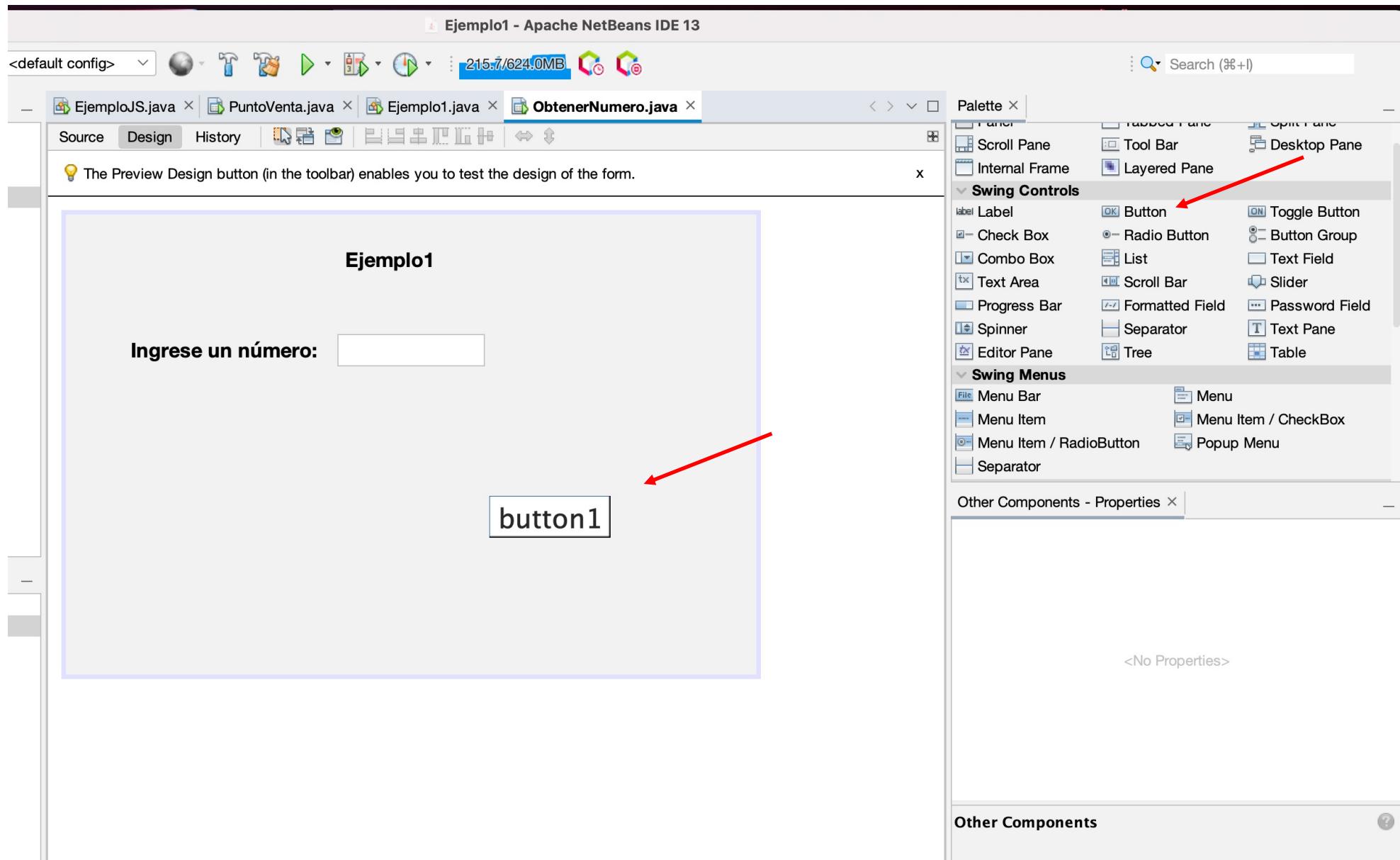
Cuando seleccionas el JLabel puedes cambiar el Texto directamente con doble click o editando con las opciones que aparecen en la parte inferior derecha



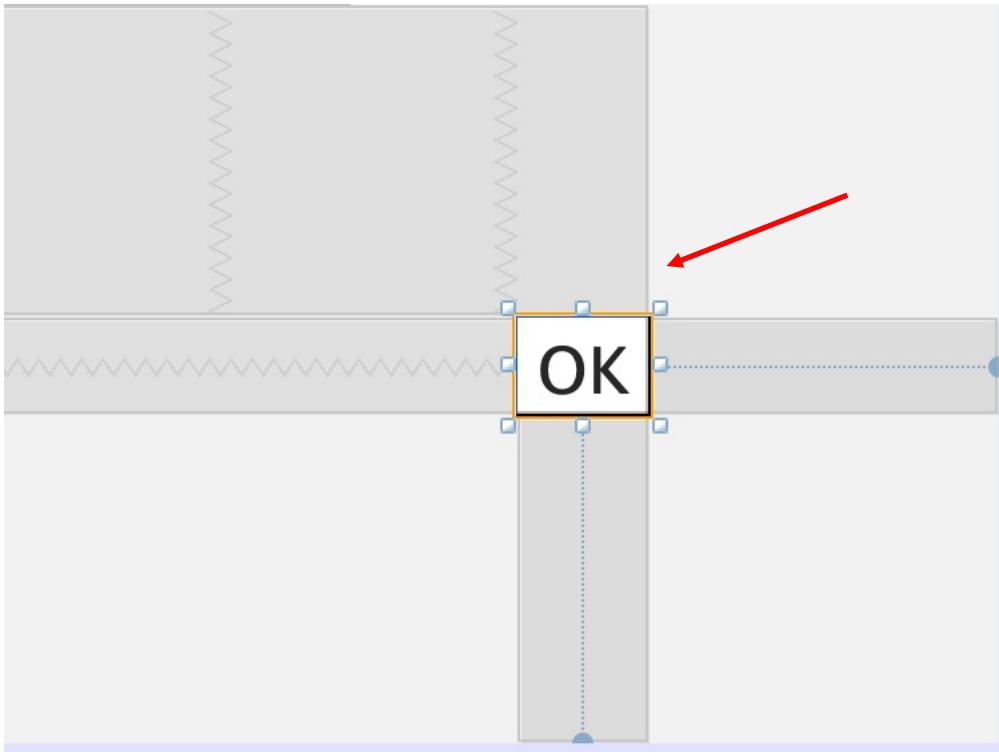
Con Text Field (jTextField) podemos crear una celda para ingresar texto o un numero.



Se puede agregar un botón con Button



Y se puede editar el texto del botón en la propiedad label. Recordar que en la propiedades podemos agrandar el botón, cambiar la letra, colocarle fondo de algún color, etc



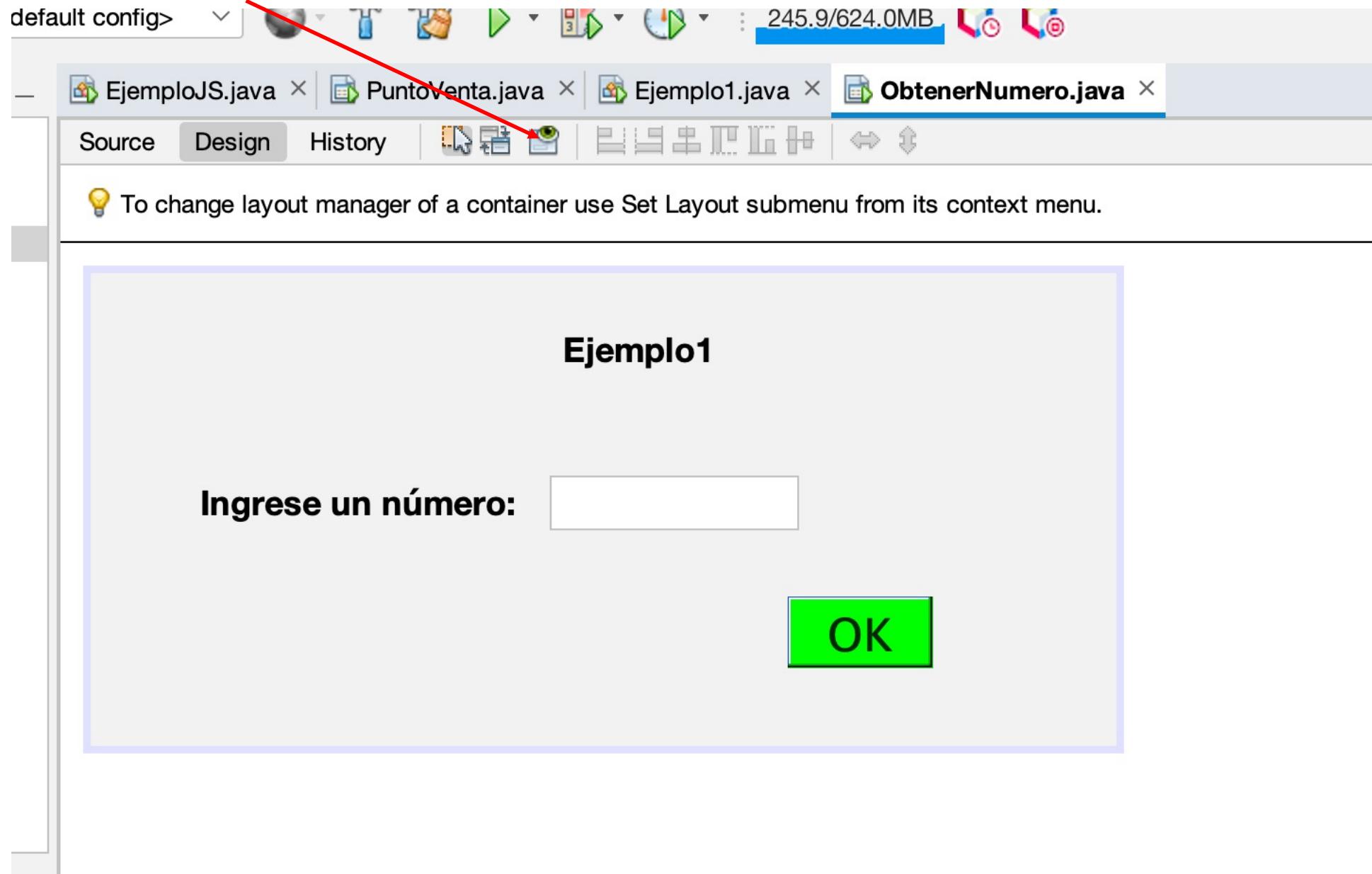
Swing Menus

- File Menu Bar
- Menu Item
- Menu Item / RadioButton
- Separator
- Menu
- Menu Item / CheckBox
- Popup Menu

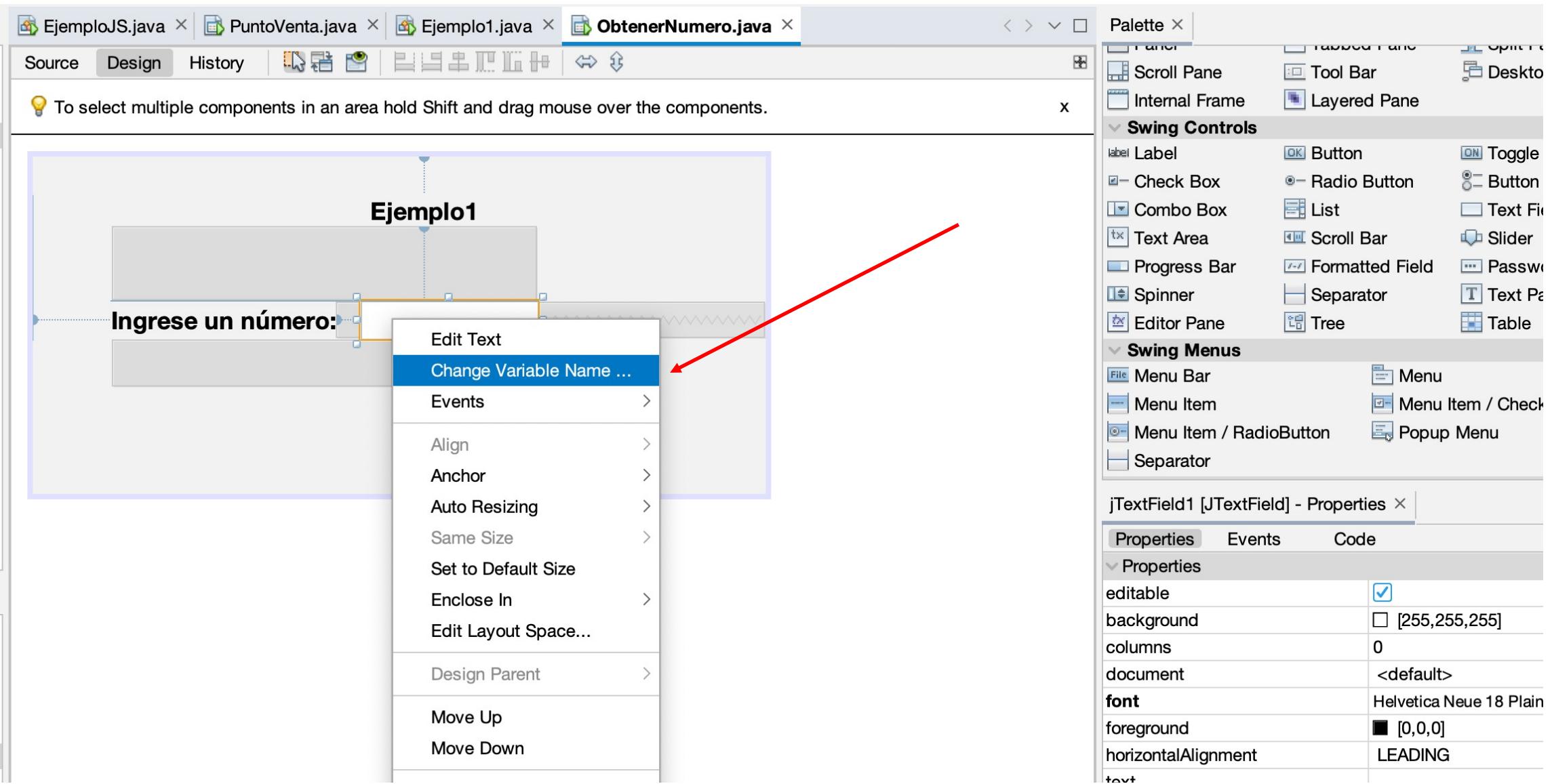
button1 [Button] - Properties

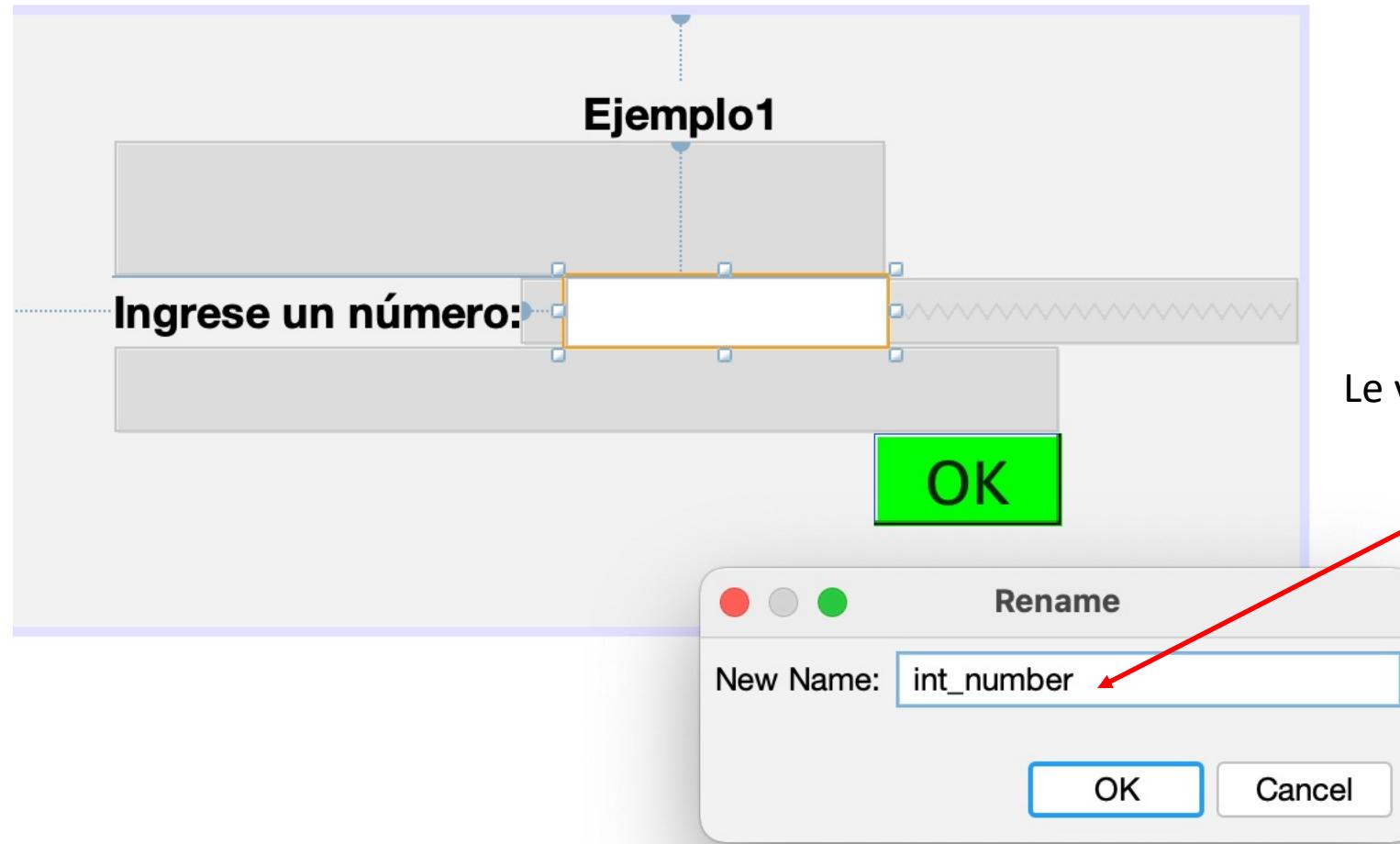
Properties	Events	Code
hierarchyListeners	<default>	...
ignoreRepaint	<input type="checkbox"/>	...
inputContext	<default>	...
inputMethodListeners	<default>	...
inputMethodRequests	<none>	...
keyListeners	<default>	...
label	OK	...
lightweight	<input type="checkbox"/>	...
location	[-32346, -32502]	...
maximumSize	[29767, 32767]	...
maximumSize... (java.awt.Point)	location	...
minimumSize	[66, 24]	...
label		...

Se puede ver como va quedando el diseño presionando ese botón



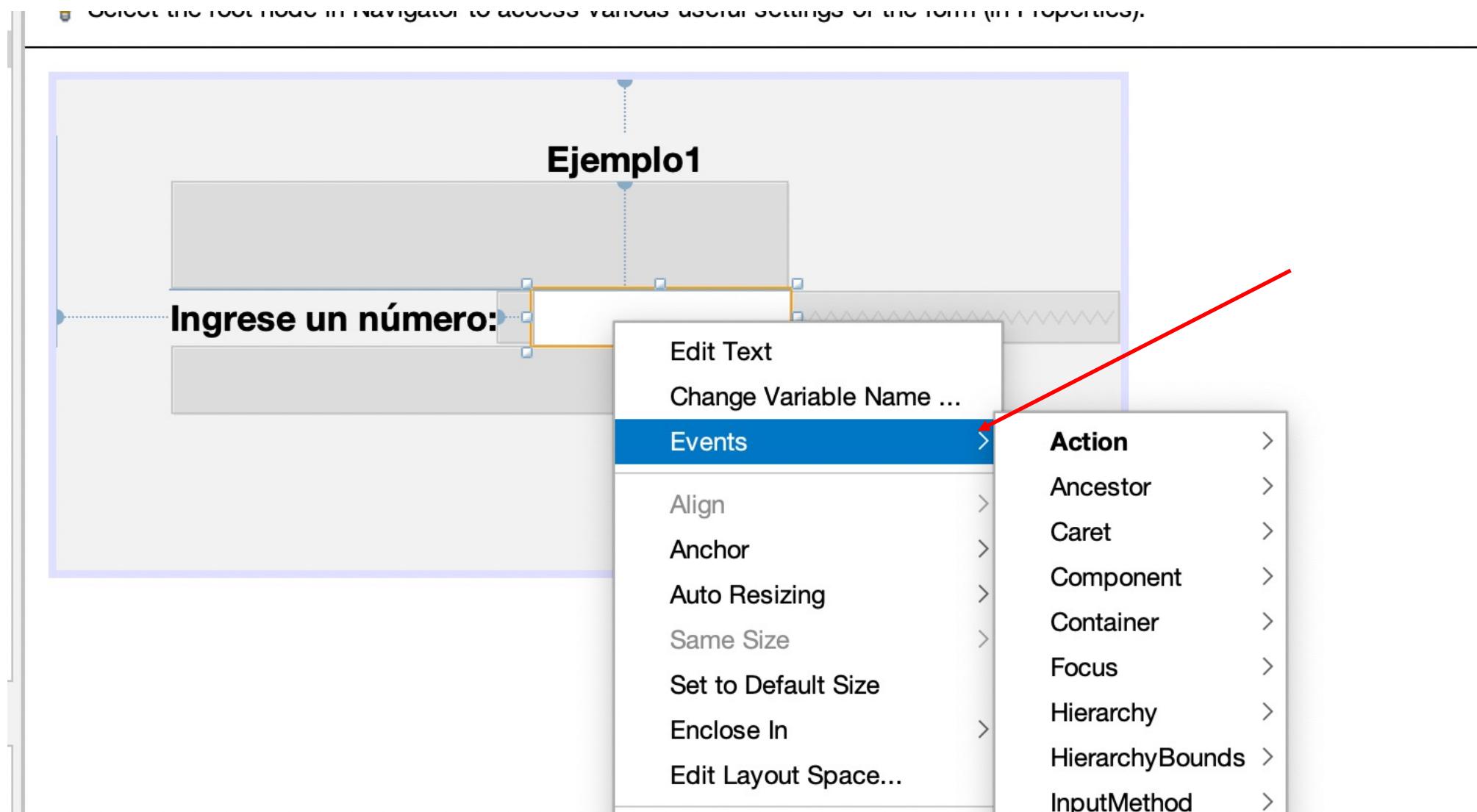
Se puede cambiar el nombre de la variable, haciendo click con el botón derecho.



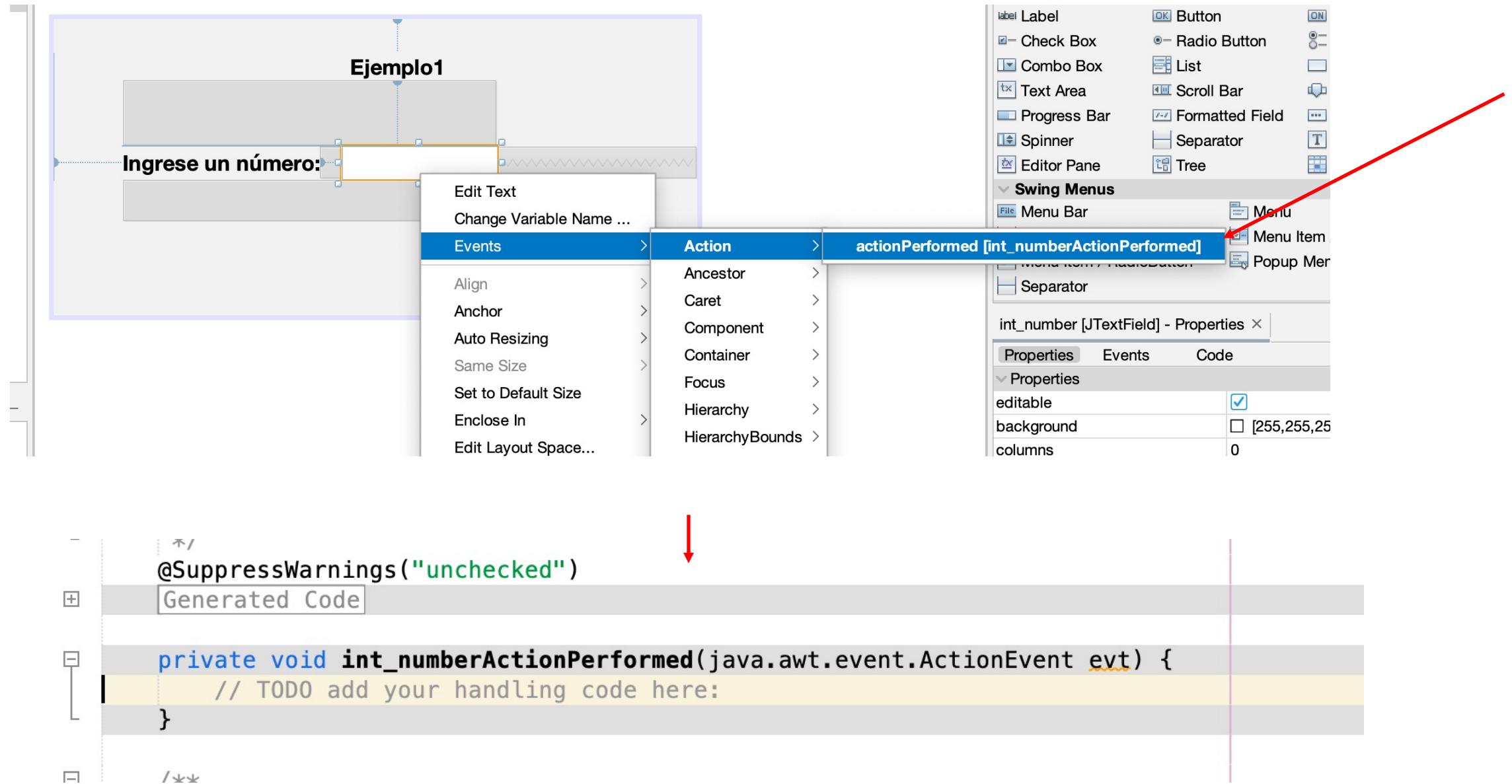


Le vamos a colocar int_number

Con el mismo botón derecho, existen otras opciones como Eventos



Si queremos añadir al código un actionPerformed, podemos hacerlo desde aquí y nos mostrará en que parte del código podemos comenzar a añadir código.



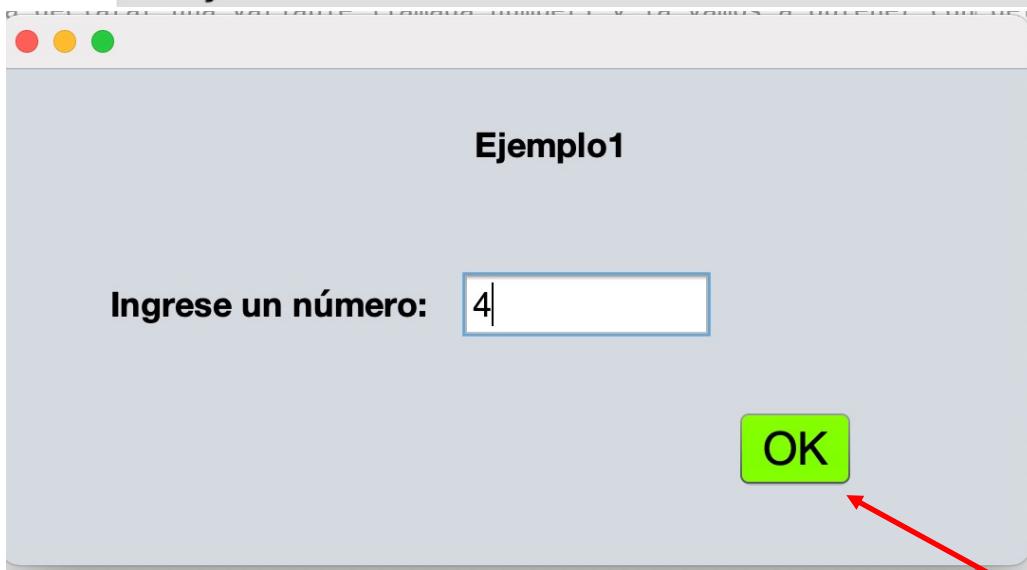
Así podría obtener un número. Recordar que se espera un string, por lo que debemos usar `Float.parseFloat()` para obtener el número.

```
private void int_numberActionPerformed(java.awt.event.ActionEvent evt) {  
    //Vamos a declarar una variable llamada number1 y la vamos a obtener con getText.  
    double number1 = 0.0;  
    //Al querer obtener un número y no un string que es lo que se espera, debemos usar en este caso ParseFloat.  
    number1 = Float.parseFloat(int_number.getText());  
}  
/*|
```

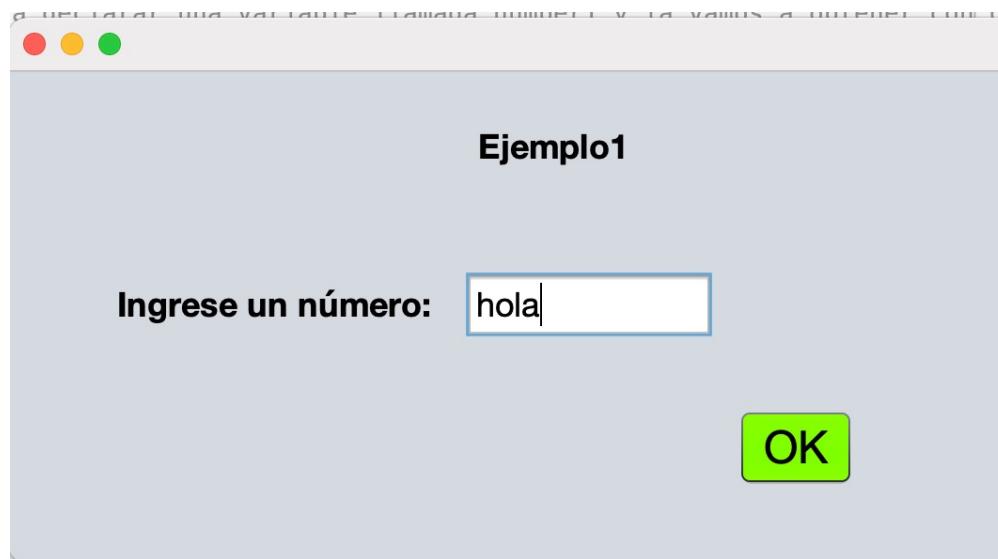
Sin embargo, piden obtenerlo y mostrarlo cuando se presiona el botón!!

Entonces....Aquí lo vamos a hacer cuando se presione el botón OK, y vamos a asumir que se ingrese un número entero:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    int number1 = 0;  
  
    number1 = Integer.parseInt(int_number.getText());  
    JOptionPane.showMessageDialog(null, "Tu número es:" + number1);  
}  
}
```



¿Qué pasa si coloco un string?



Fijarse en el error!

```
Output - Ejemplo1 (run) ×
Exception in thread "AWT-EventQueue-0" java.lang.NumberFormatException: For input string: "hola"
    at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
    at java.base/java.lang.Integer.parseInt(Integer.java:668)
    at java.base/java.lang.Integer.parseInt(Integer.java:784)
    at ejempl01.ObtenerNumero.jButton1ActionPerformed(ObtenerNumero.java:102)
```

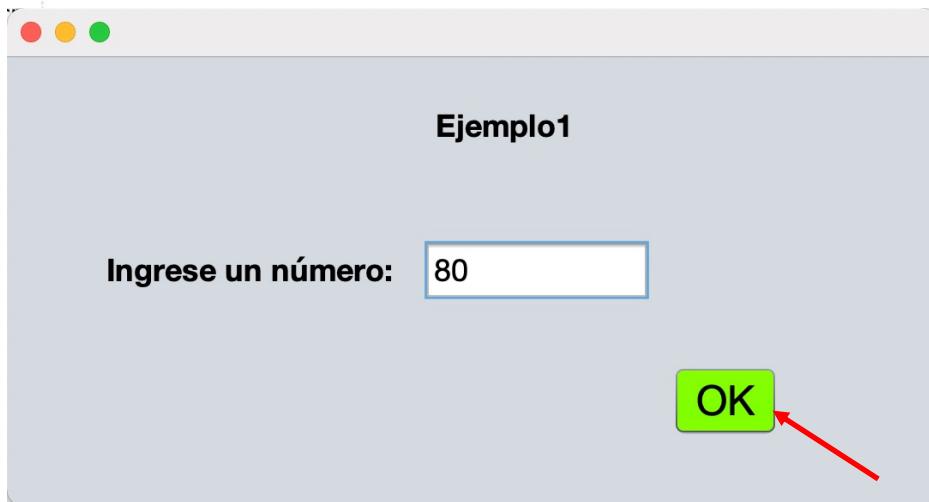
Podemos arreglar esto con try – catch y como el error era “NumberFormatException” lo podemos hacer así:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    int number1 = 0;  
    try{  
        number1 = Integer.parseInt(int_number.getText());  
        JOptionPane.showMessageDialog(null, "Tu número es:" + number1);  
    }catch(NumberFormatException nfe){  
        JOptionPane.showMessageDialog(null, "Sólo se aceptan valores numéricos");  
    }  
}
```



Ejemplo 2: Lo mismo pero que muestre y reciba Float

```
private void int_numberActionPerformed(java.awt.event.ActionEvent evt) {  
}  
  
}  
  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    double number1 = 0.0;  
    try{  
        number1 = Float.parseFloat(int_number.getText());  
        JOptionPane.showMessageDialog(null, "Tu número es:" + number1);  
    }catch(NumberFormatException nfe){  
        JOptionPane.showMessageDialog(null, "Sólo se aceptan valores numéricos");  
    }  
}
```

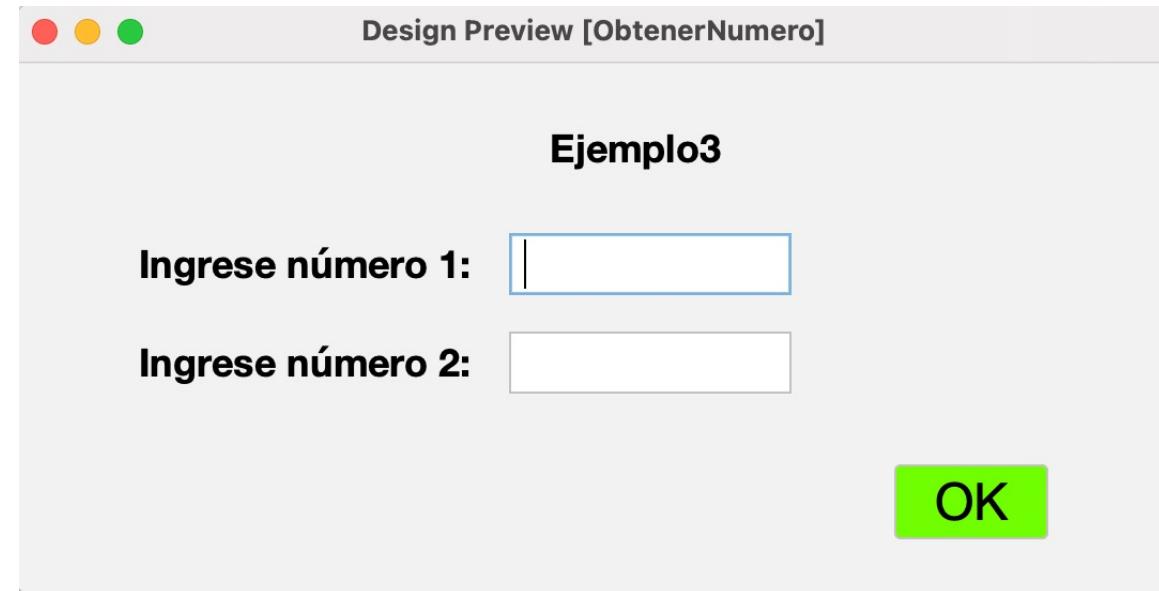


Agregaremos una condición, si el número ingresado por el usuario es mayor que 50, mostraremos un mensaje que diga: "tu número es mayor a 50!" de lo contrario que continue mostrando el mensaje de antes

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    double number1 = 0.0;  
    try{  
        number1 = Float.parseFloat(int_number.getText());  
  
        if (number1 > 50){  
            JOptionPane.showMessageDialog(null, "Tu número es mayor a 50!");  
        }else{  
            JOptionPane.showMessageDialog(null, "Tu número es :" + number1);  
        }  
    }catch(NumberFormatException nfe){  
        JOptionPane.showMessageDialog(null, "Sólo se aceptan valores numéricos");  
    }  
}
```

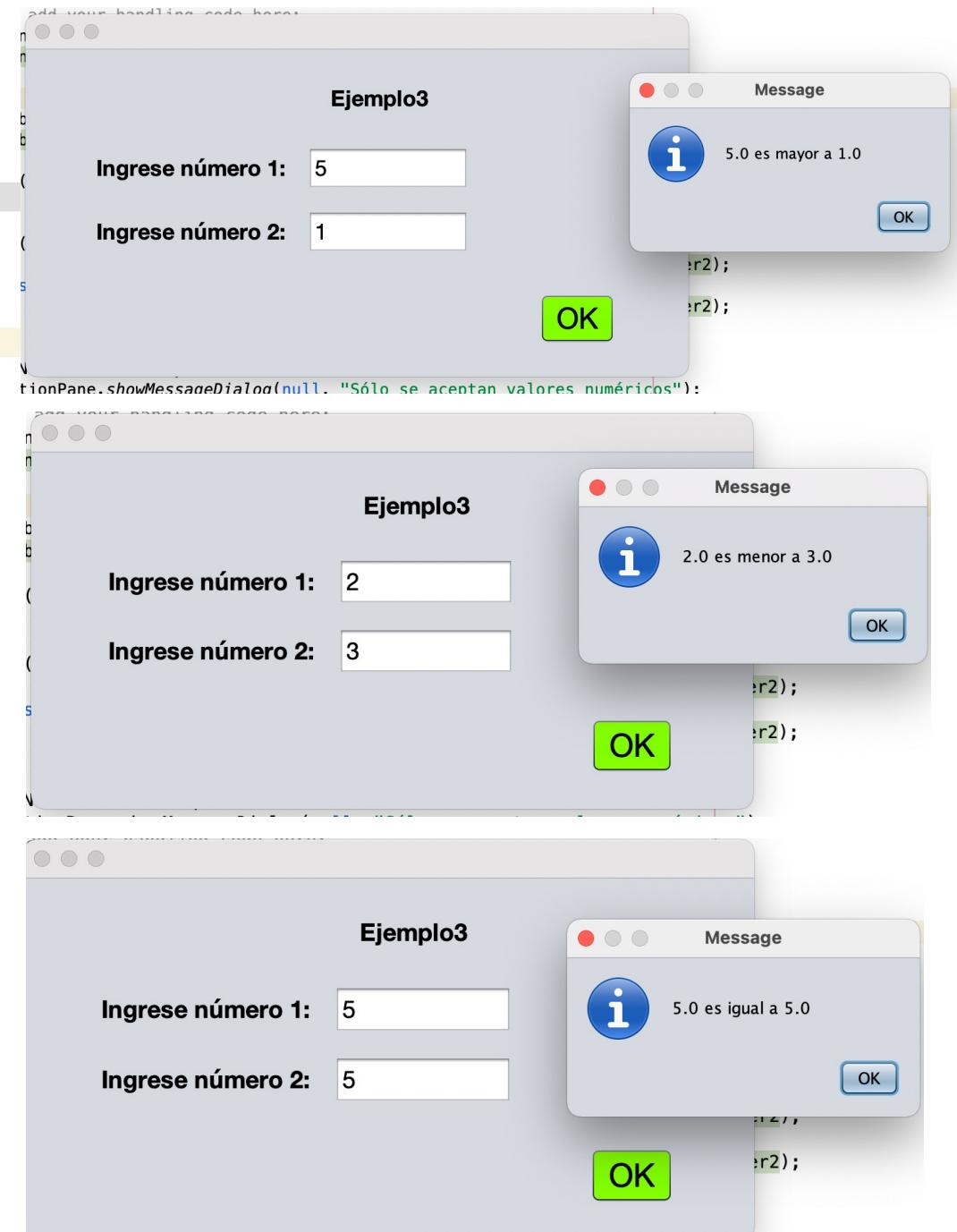


Ejemplo3: Que el usuario ingrese dos números y que sea capaz de decir si el primer numero es mayor, menor o igual a segundo número.

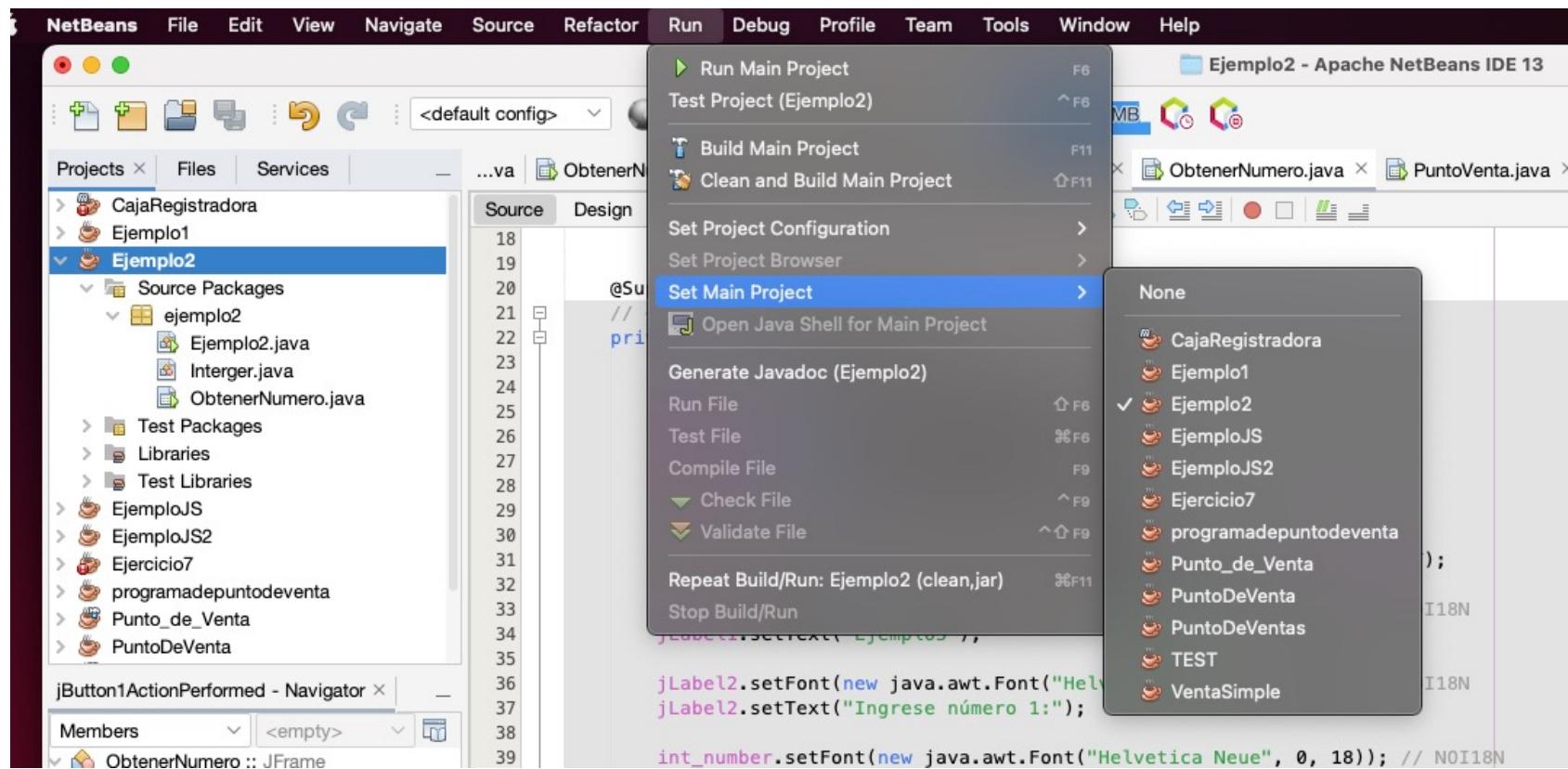


Ejemplo3

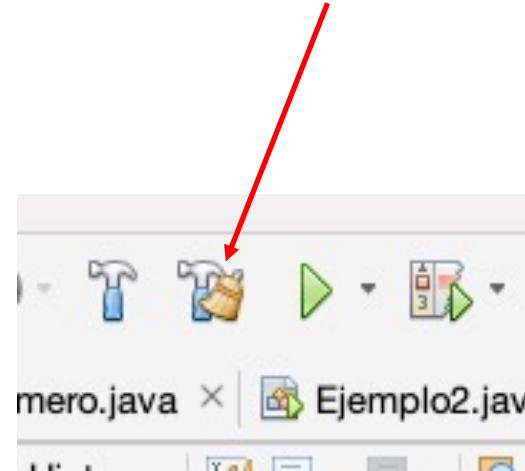
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    double number1 = 0.0;  
    double number2 = 0.0;  
  
    try{  
        number1 = Float.parseFloat(int_number.getText());  
        number2 = Float.parseFloat(int_number2.getText());  
  
        if (number1 > number2){  
            JOptionPane.showMessageDialog(null, number1 + " es mayor a " + number2);  
        }  
        if (number1 < number2){  
            JOptionPane.showMessageDialog(null, number1 + " es menor a " + number2);  
        }else{  
            JOptionPane.showMessageDialog(null, number1 + " es igual a " + number2);  
        }  
  
    }catch(NumberFormatException nfe){  
        JOptionPane.showMessageDialog(null, "Sólo se aceptan valores numéricos");  
    }  
}
```



- Para guardar deben ir a:
- Menu → Run → set Main Project → Selecciona el proyecto que estas trabajando o que deseas enviar (en este caso Ejemplo2).



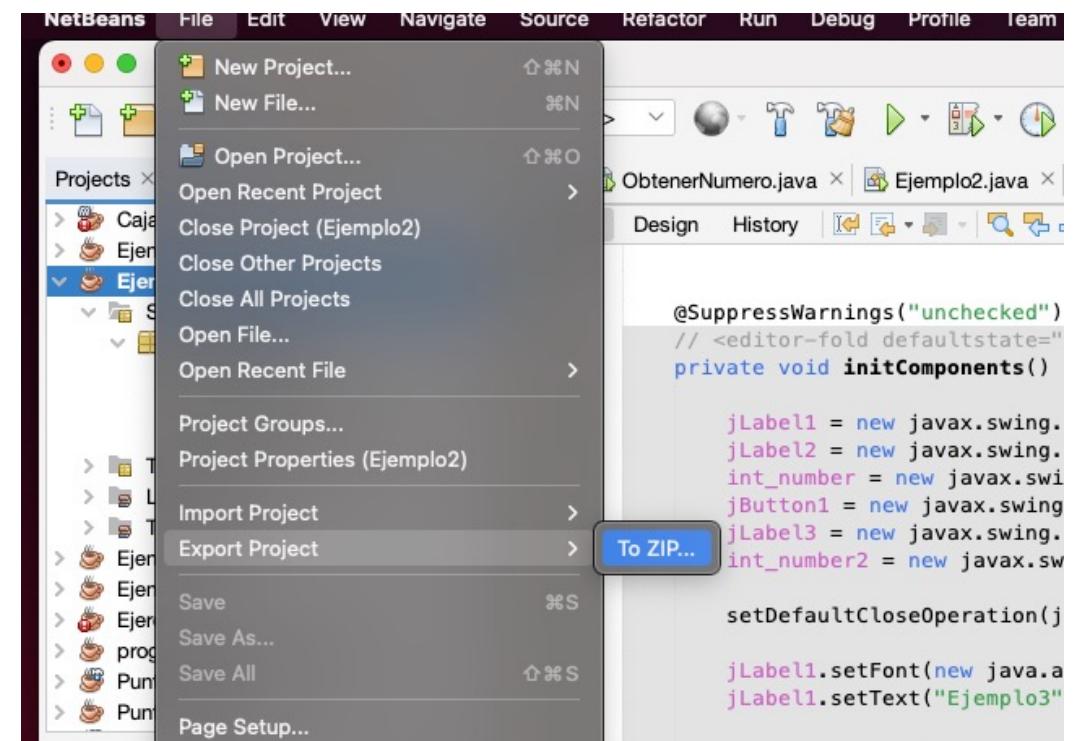
- Luego recomiendo colocar Clean and Build



Y finalmente en el menú existe la opción para exportar a .zip

Menú → Export Project → To Zip....

Para saber si realmente se guarda correctamente, intenten cargar el proyecto ustedes a NetBeans con la opción de "Import Project"



¿Qué se pide en este ejercicio? Algo como esto (sólo una idea):

The screenshot shows a software window with the following interface elements:

- Vendedor:** Andrea Castillo (dropdown menu)
- Ingrese Monto de Venta \$:** (Text input field)
- ¿Aplica Descuento?** (Button)
- Porcentaje de Descuento %:** (Text input field)
- TOTAL:** (Text input field)
- Pagar** (Button)

- 1.- Se ingresa un monto de venta
Si ingresamos 1000 no debería aplicarse el descuento
Si es 1001 debería tener despuesto de 20%
Mostrar total a pagar de forma correcta.

Vendedor: Andrea Castillo

Ingrese Monto de Venta \$: 1000

¿Aplica Descuento?

Porcentaje de Descuento %: 0.0

TOTAL: 1000.0

Pagar

Vendedor: Andrea Castillo

Ingrese Monto de Venta \$: 1200

¿Aplica Descuento?

Porcentaje de Descuento %: 20.0

TOTAL: 960.0

Pagar