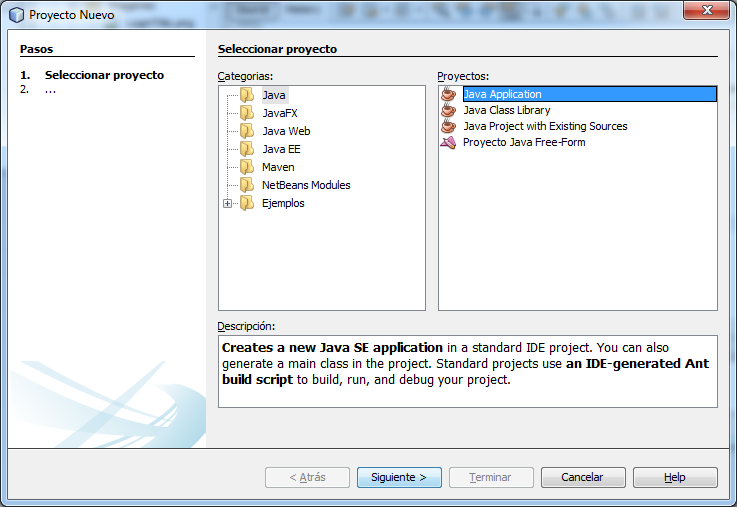
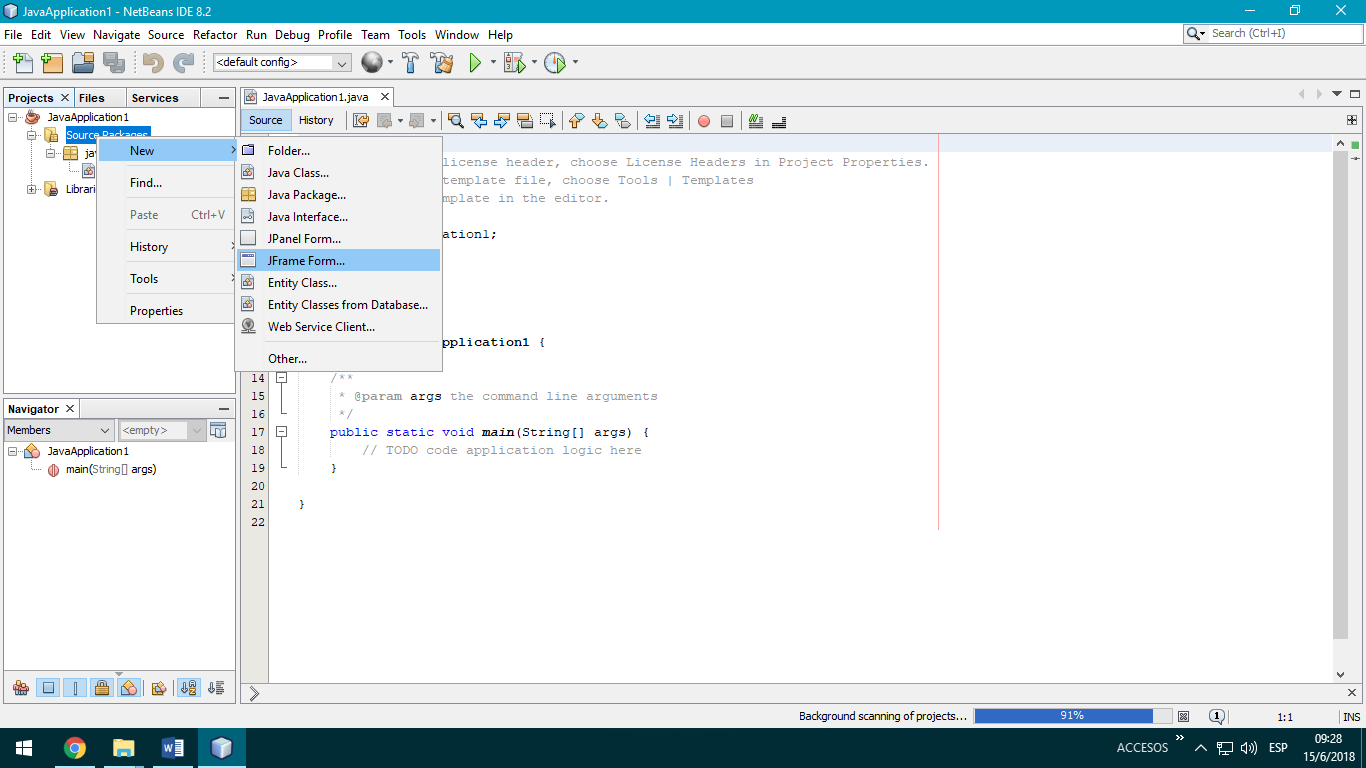
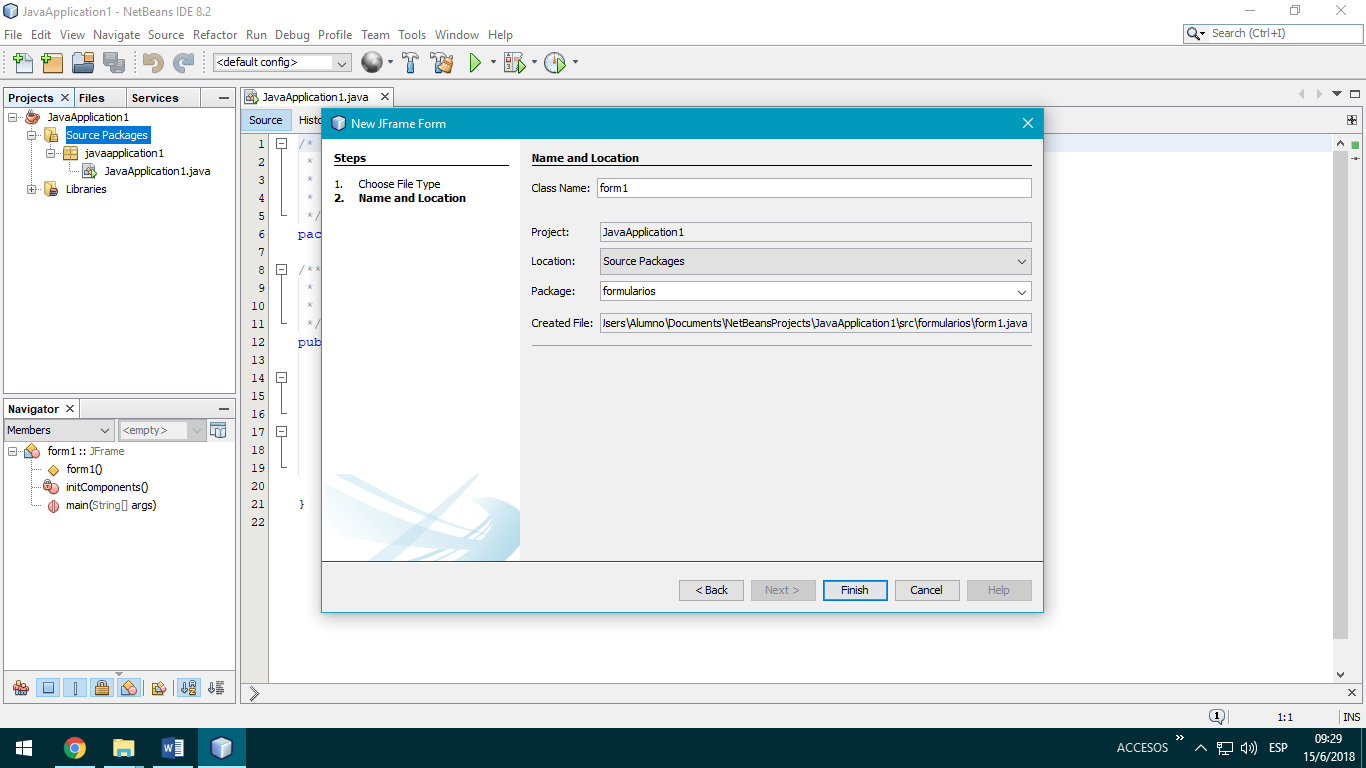
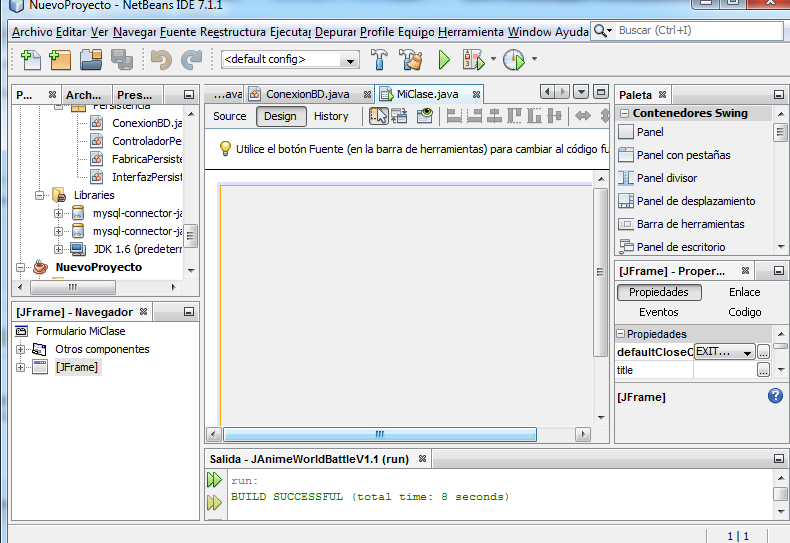
**Interfaz visual**

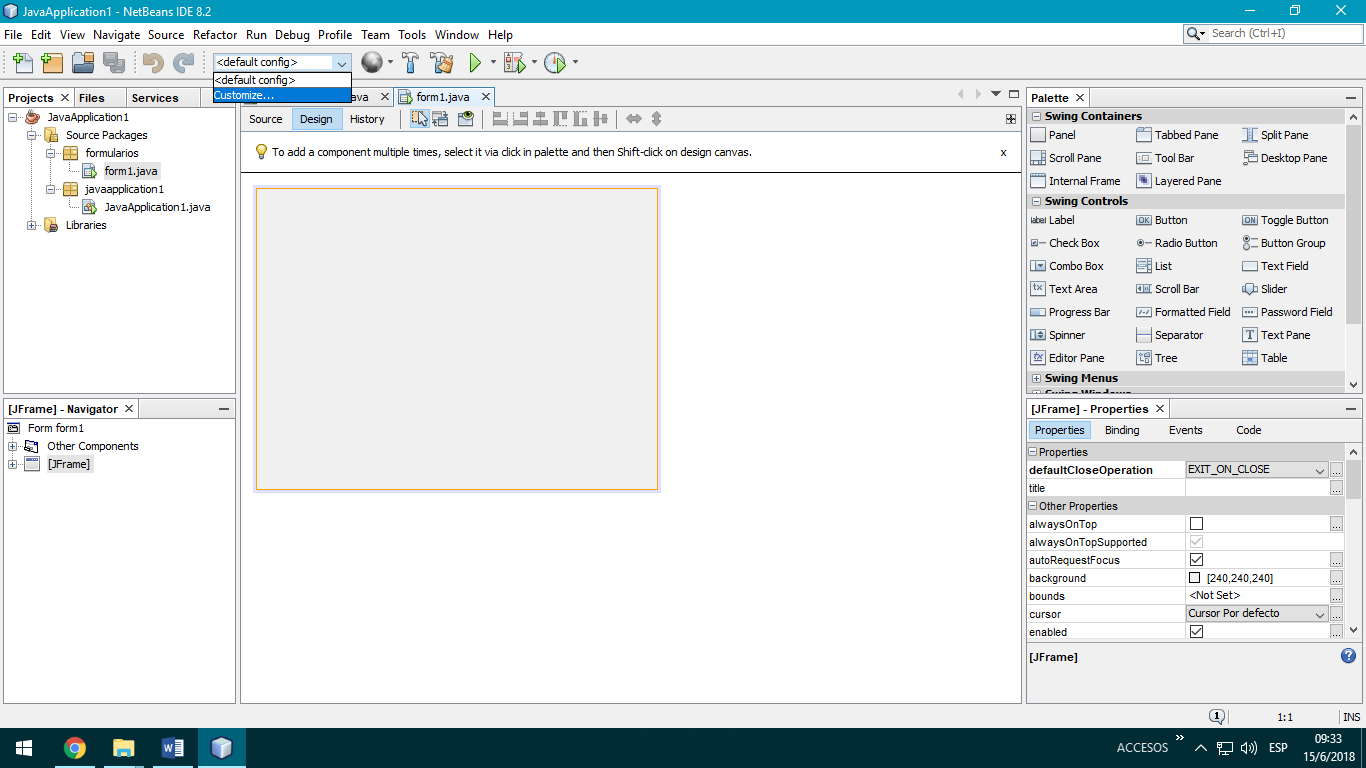
Nos creamos un nuevo proyecto, aquí le pongo de nombre**NuevoProyecto**. Lo elegimos como **Java Aplication**  
  
  
En mi caso no le he creado clase principal porque por lo general me gusta crearla yo mismo. Más allá de eso agregaremos una clase (que puede servir como principal) que te permita trabajar tal como si fuera una Desktop Aplication, así:  
  
Haces clic en Source package --> Nuevo --> Formulario JFrame.  
  


Completar el nombre y el package con el nombre formularios, esto crea una carpeta formularios donde guardaremos todas las ventanas o jframe

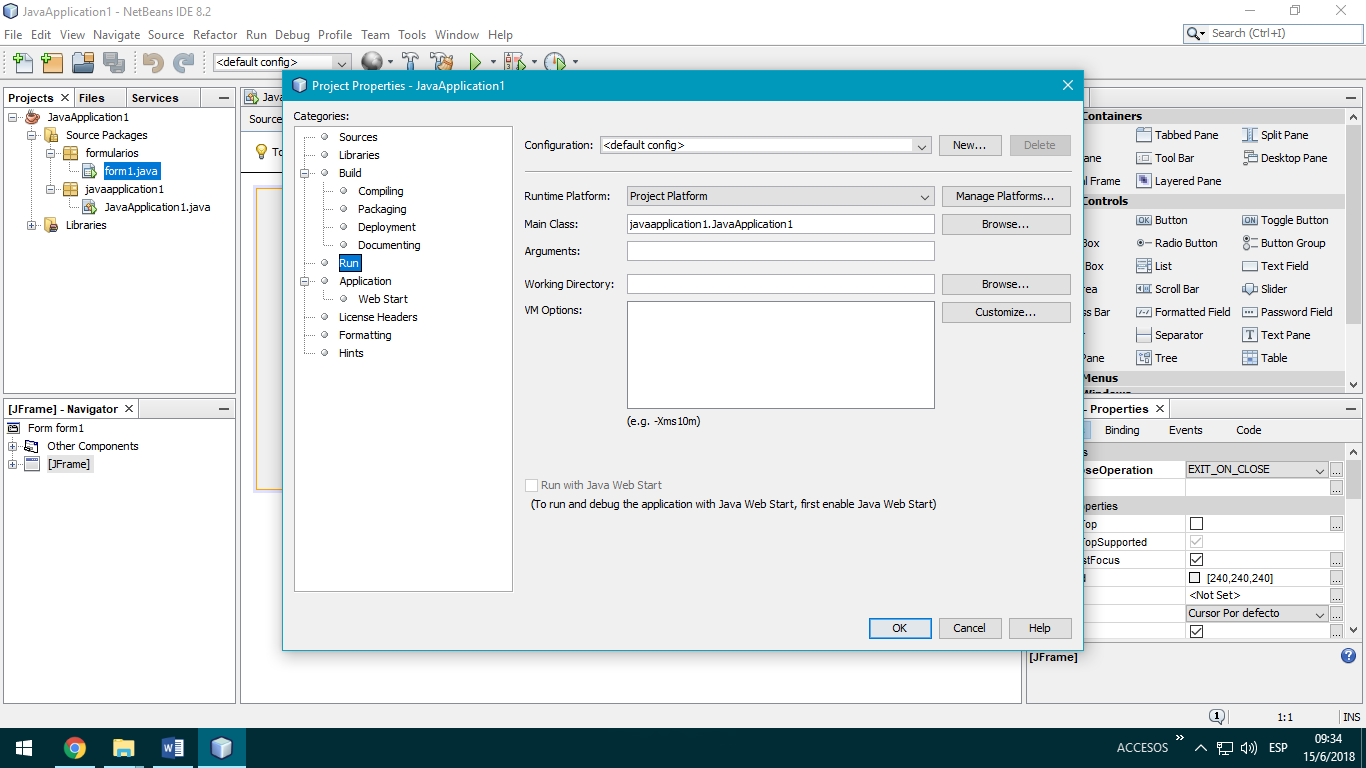


Como ves, NetBeans nos abre la paleta de diseño automáticamente  


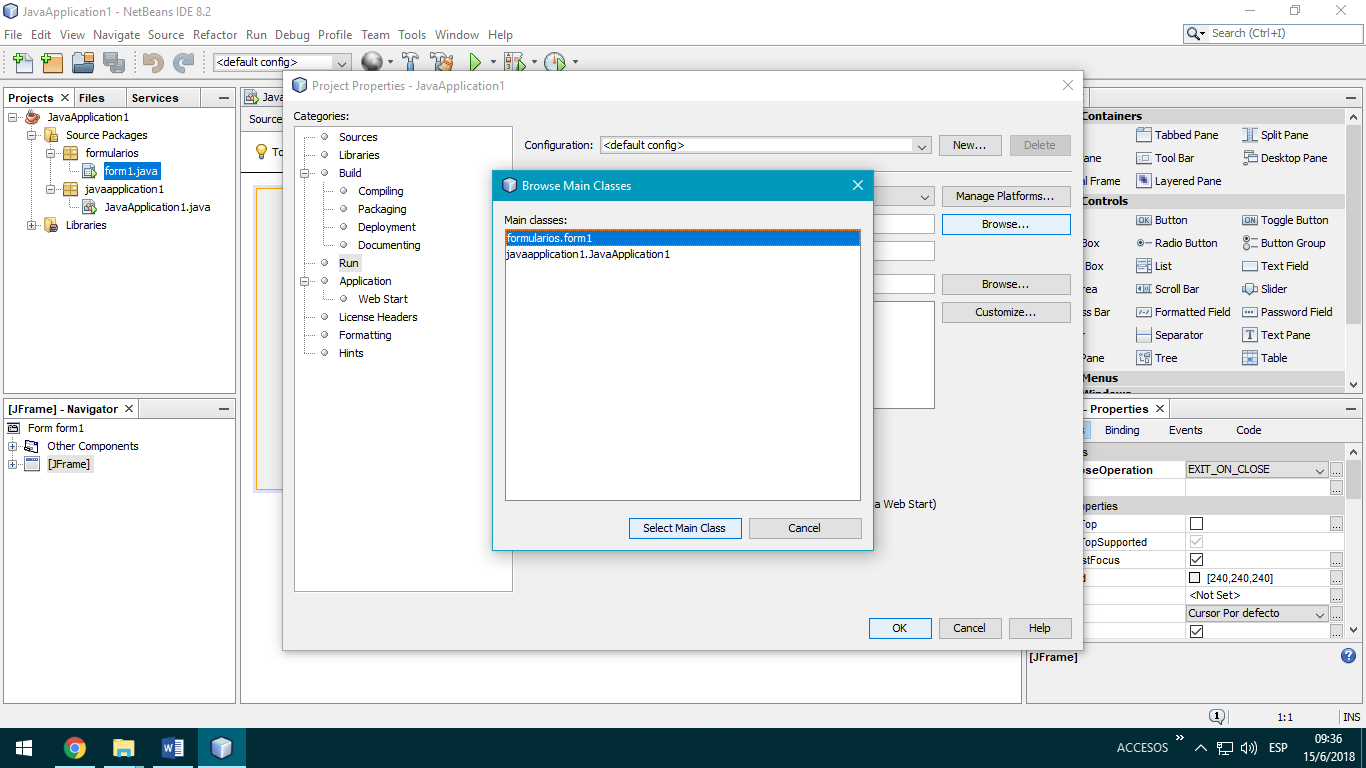
Si ejecutas, vas a vr que ejecuta la mainc donde nosotros escribíamos código, para unir el proyecto a nuestra ventana de trabajo, vamos a



Se abre y donde dice Main class, vamos a buscar con el botón Browser nuestro form



Elegir el form que queremos cmoo inicial



Ahora para que ejecute la ventana

**Jframe (formulario) ventana**

**Propiedades:**

Title: titulo de la ventana

Bounds: posición xey es la esquina izq superior y width y height ancho y alto

Probar con 200, 200. 600 de ancho y 1000 de alto

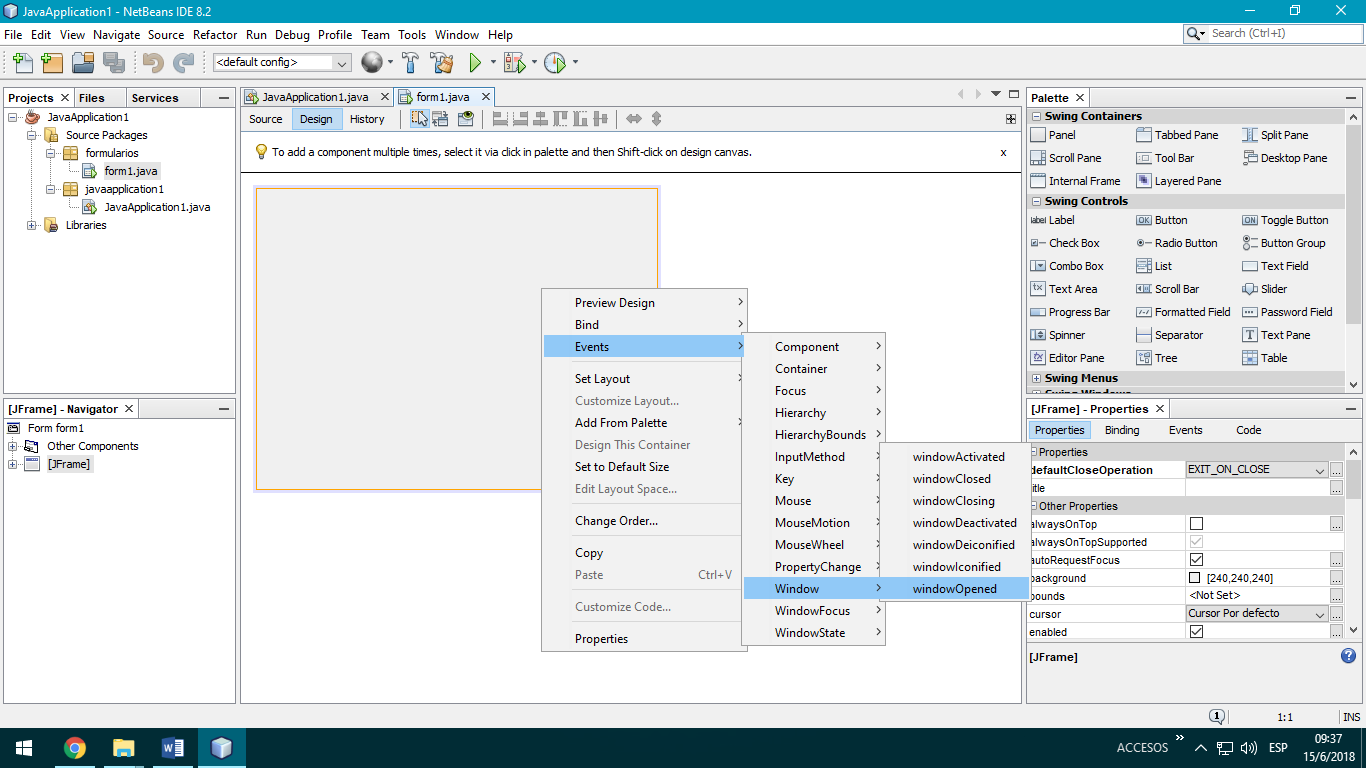
Cursor el cursor sobre la ventana

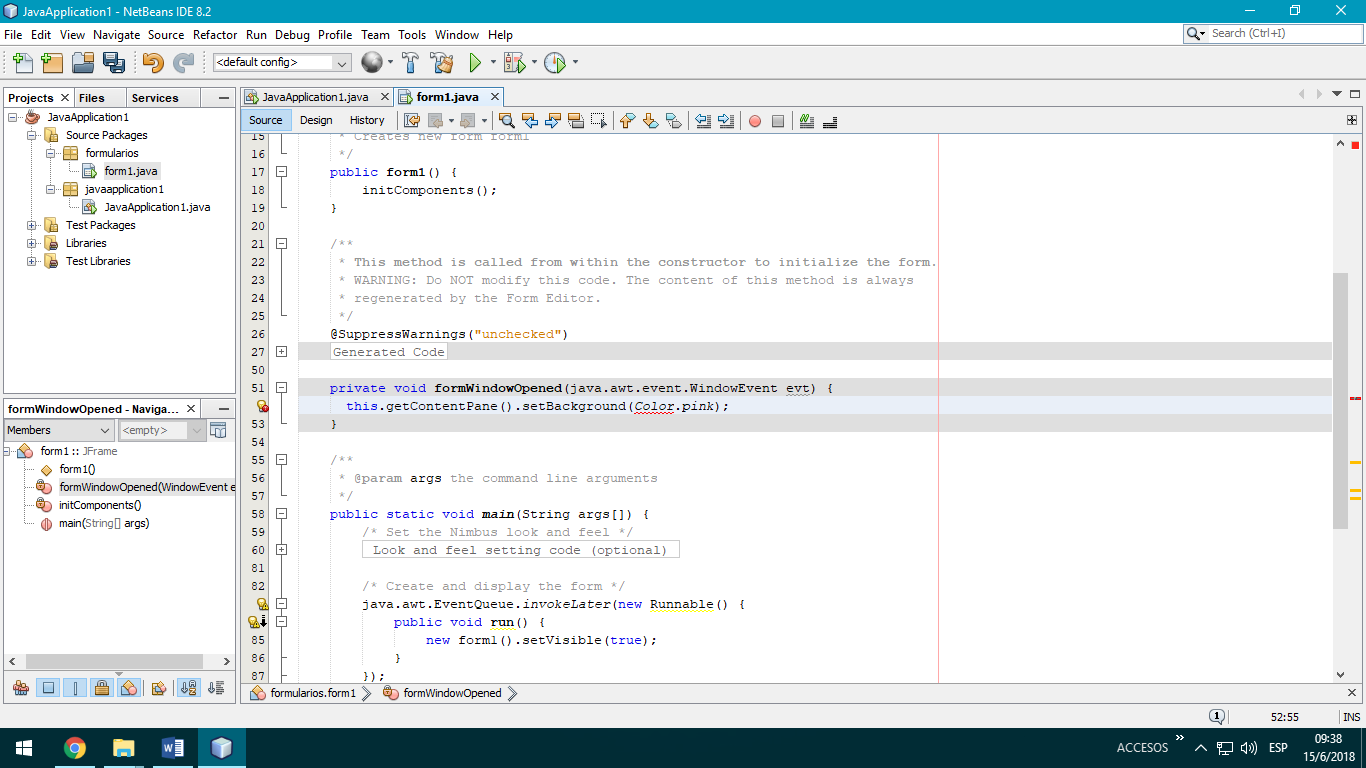
**Eventos Windowactivated Se repite constantemente**

**WindowOpened una sola vez**

Para cambiar el cursor en el Jframe

Click botón drecho Events / window / WindowOpened





this.getContentPane().setBackground(Color.pink);

Para cambiar el icono se necesita código, para cambiar el color de fondo hay que usar un panel o una jlabel

**Label etiquetas**

Text:texto que aparece en la etiqueta

Font: fuentes

Background:color fondo se debe cambiar **opaque** a true o activarla

Foreground:color de letra

Border:borde

Cursor punter del mouse

Horizontalaligmenet o vertical alignement es la aliacion

Icon: permite cargar una imagen

**Botones**

Text:texto que aparece en la etiqueta

Font: fuentes

Background:color fondo

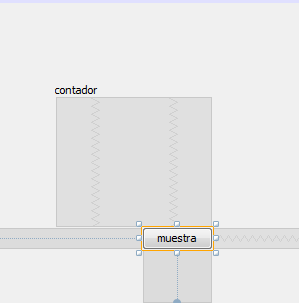
Foreground:color de letra

Border:borde

Cursor punter del mouse

Horizontaltextposition alineacion del texto con respecto aun icono.

Icon: permite cargar una imagen



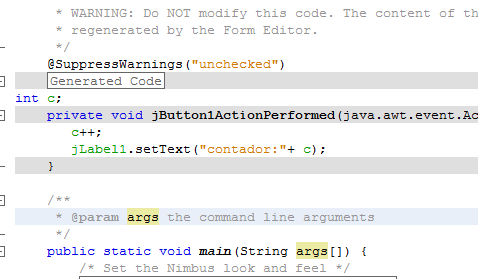
**Trabajo de propiedades desde código**

Para editar una propiedad get

Para cambiar una propiedad set

Ejemplo

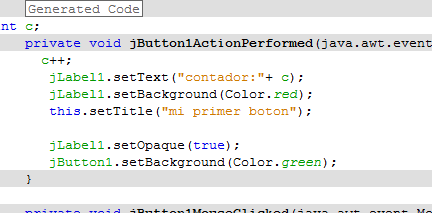
Una etiqueta y un botón



Desde este botón puedo trabajar propiedades de otros, por ejemplo:

Agregar debajo en el botón si quiero cambiar el color de fondo de la etiqueta

jLabel1.setbacground(color.red); y hay que agregar

jLabel1.setOpaque(true); porque por defecto es transparente,

Para cambiar el titulo a la ventana

this.setTitle("mi primer boton");

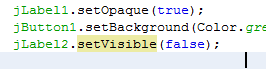
para cambiar el color del boton

jButton1.setBackground(Color.green);

agregar otro botón de salir

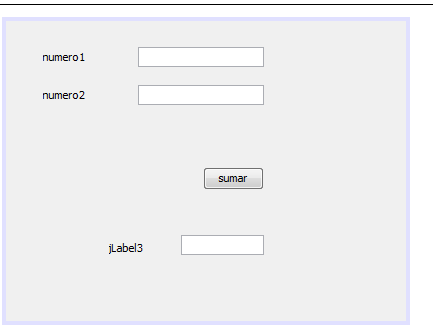
y en el código System,exit(0);

Agregar una nueva label con texto ver y en el código del botón muestra agregar al fondo



**Cuadros de texto textfield**

La propiedad mas importante es la propiedad text que es el contenido del cuadro



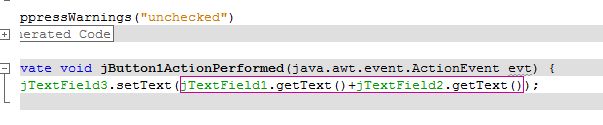
Background

Foreground

Font

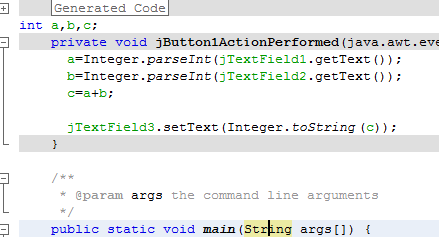
Editable: si quieren quitar el cursor

Código del botón



**Si probamos esto, no funciona, concatena el contenido de los cuadros**

Para que funcione bien, debemos usar variables



De esta forma pueden tomar el contenido de cualquier texto

Para textbox: c=Integer.parseInt(jTextField1.getText());

Para convertir a string string

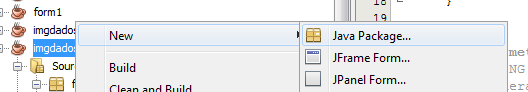
integer.toString(variable)

Crear el jar y probarlo

**Imágenes**

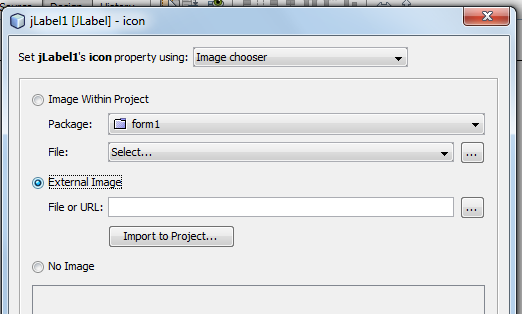
Para cargar imágenes vamos a usar jlabel

Lo primero que vamos a hacer es crear una carpeta de imagnes para guardarls



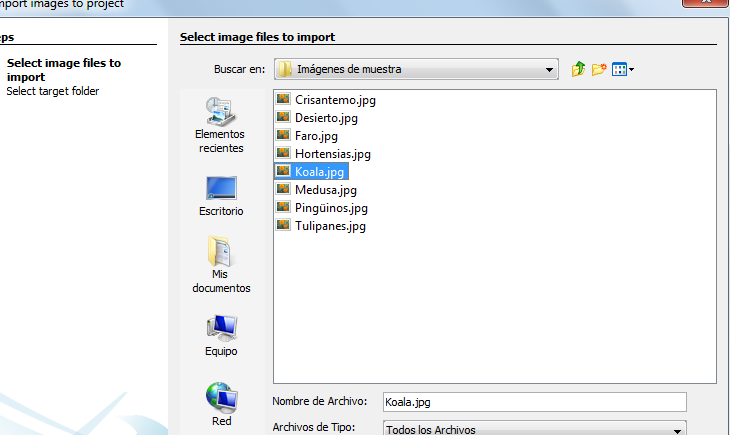
Y vamos a guarcar todas las imagnes en esa carpeta

Propiedad icon

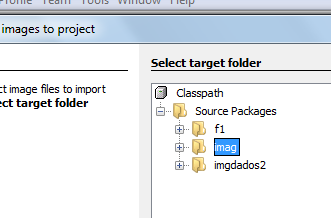


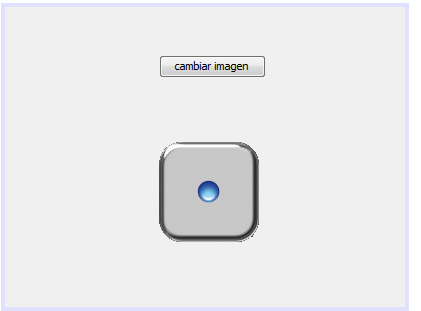
buscar e importar

Una vez seleccionada importar



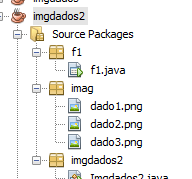
Donde se guarda y finalizar (usar la carteta de imágenes)





Crear un proyecto con label y agregar un botón

**Nota: copiar y pegar las imagenes en la carpeta imag**

****

En el botón

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

 jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/imag/dado3.png")));

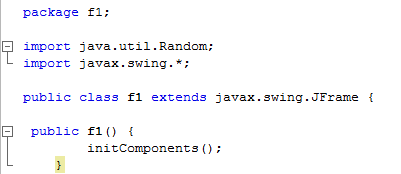
}

Agregar una label arriba con el gif animado y el texto del boton por tirar dado y probar

Import el random

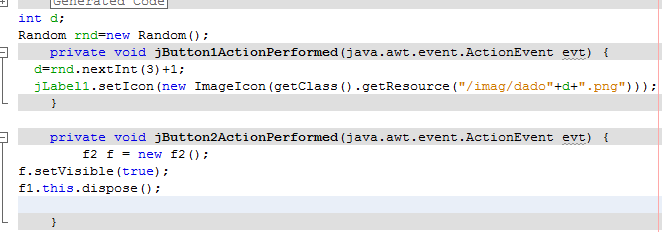
Y el swing

Import java.swing.\*;



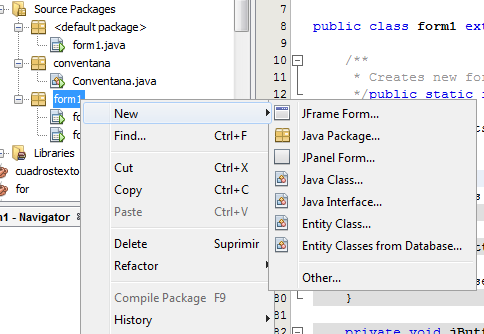
Y en el botón

Generar el valor y cargarlo



**Multiples ventanas**

Para crear mas de una ventana se puede trabajar con jpanel o jframe

Agregar un nuevo jframe y en un botón (donde f2 es el nombre del jframe)

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

f2 fr = new f2();

fr.setVisible(true);

f1.this.dispose();

}

Agregar en el jframe2 un botón anterior el jframe se llama f2

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

f1 fr = new f1();

fr.setVisible(true);

f2.this.dispose();

}

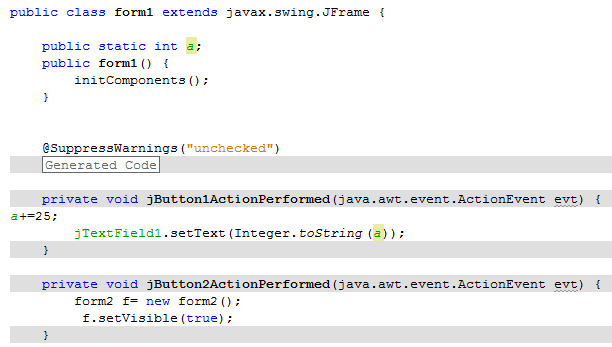
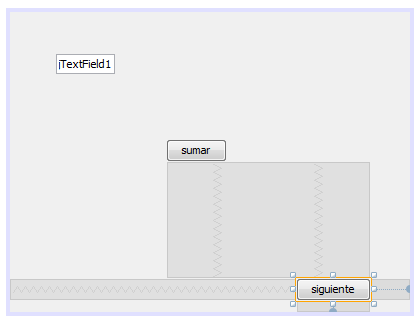
**Variables globales y locales**

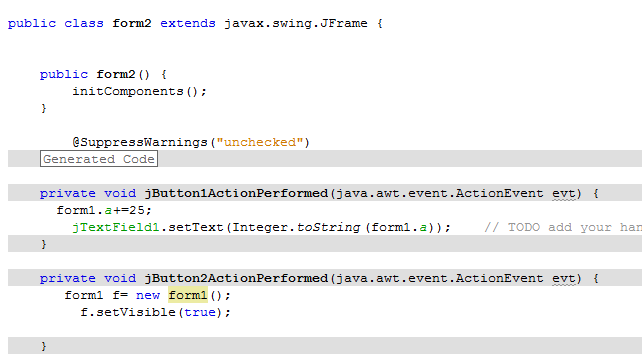
Agregar nuevo form

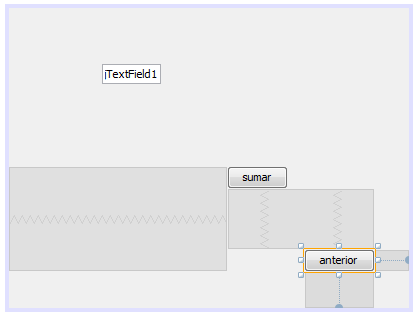
Dentro de la clase form1, agregar un form2

En el form1

Definir como public static y para llamarla form1.nombre de variable







Si se tiene el mismo código en 2 forms, se puede crear una función

Se la crea con public static nombre(){

}

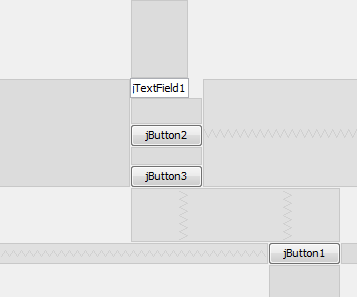
Desde un botón se puede llamar a la public starttic (no se puede trabajr objetos, solo es de variables y todas deben ser public static

Agregamos un botón 3

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

form2 f2=new form2();

f2.setVisible(true);



}

public static int c;

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

c++;

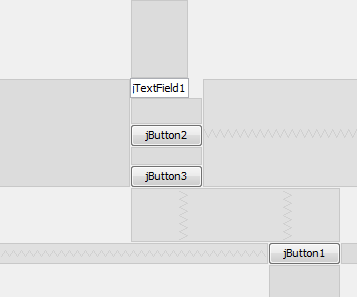
jTextField1.setText(""+c);

}

public static void suma (){

c++;

}

 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

suma();

jTextField1.setText(""+c);

}

Y en form2

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

form1 f1=new form1();

f1.setVisible(true);

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

form1.c++;

jTextField1.setText(""+form1.c);

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

form1.suma();

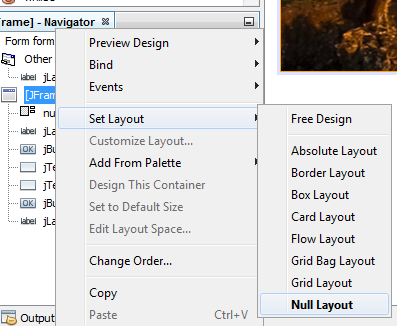
jTextField1.setText(""+form1.c);

}

**Imagen de fondo (forma fácil)**

**Nota importante: esto hacerlo una vez finalizado el proyecto, no agregar controles luego**

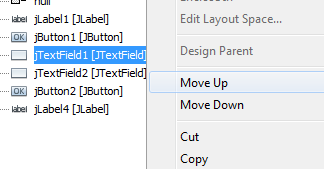
Hacer todo el proyecto **al finalizar**



Poner una label cargarle la imagen de fondo y volver a habilitar free Design (si no lo cambio queda pequeño)

No importa que no se vean algunos objetos en diseño, se ven en ejecución

Si se agrega algún objeto luego usar

traer al frente o atrás