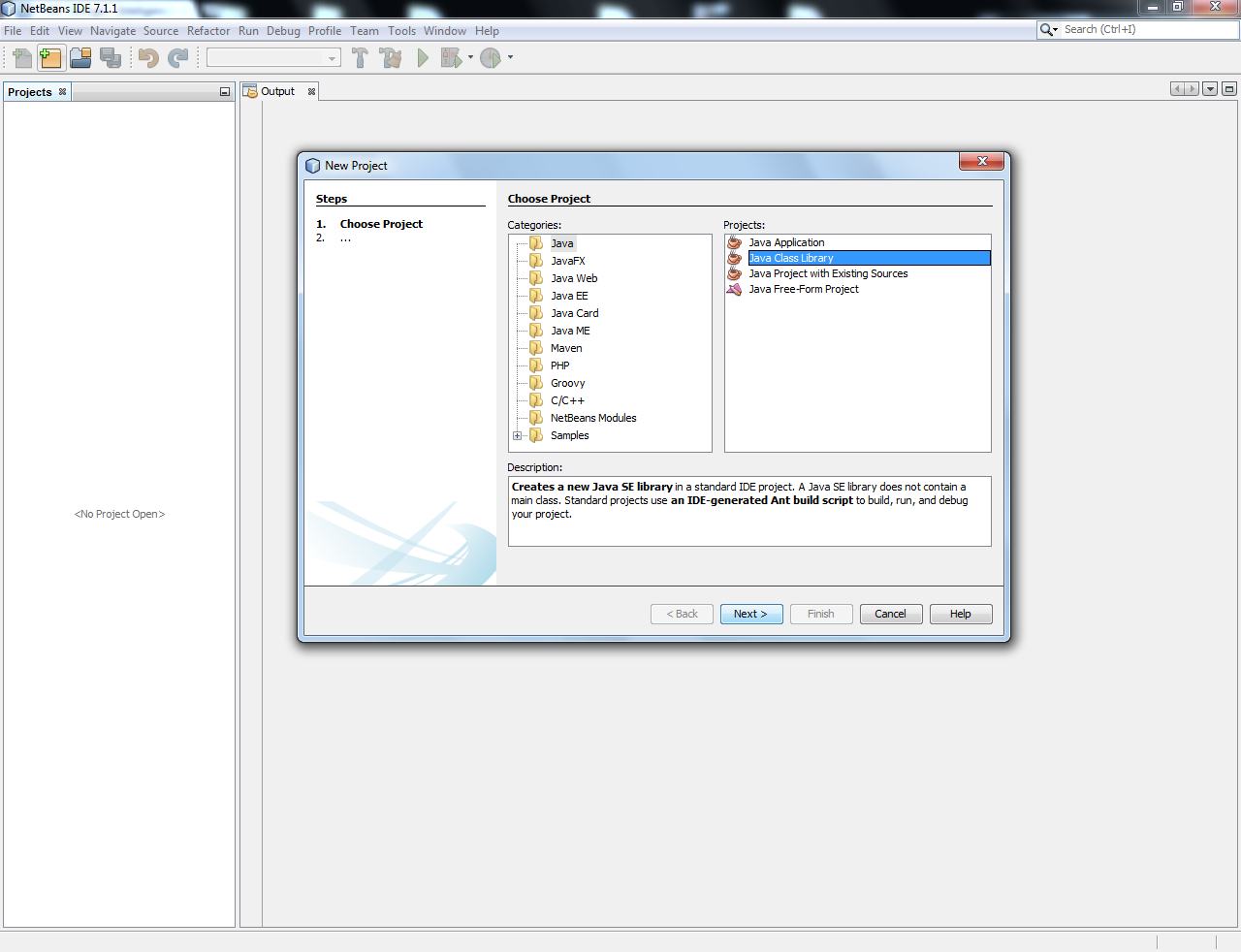
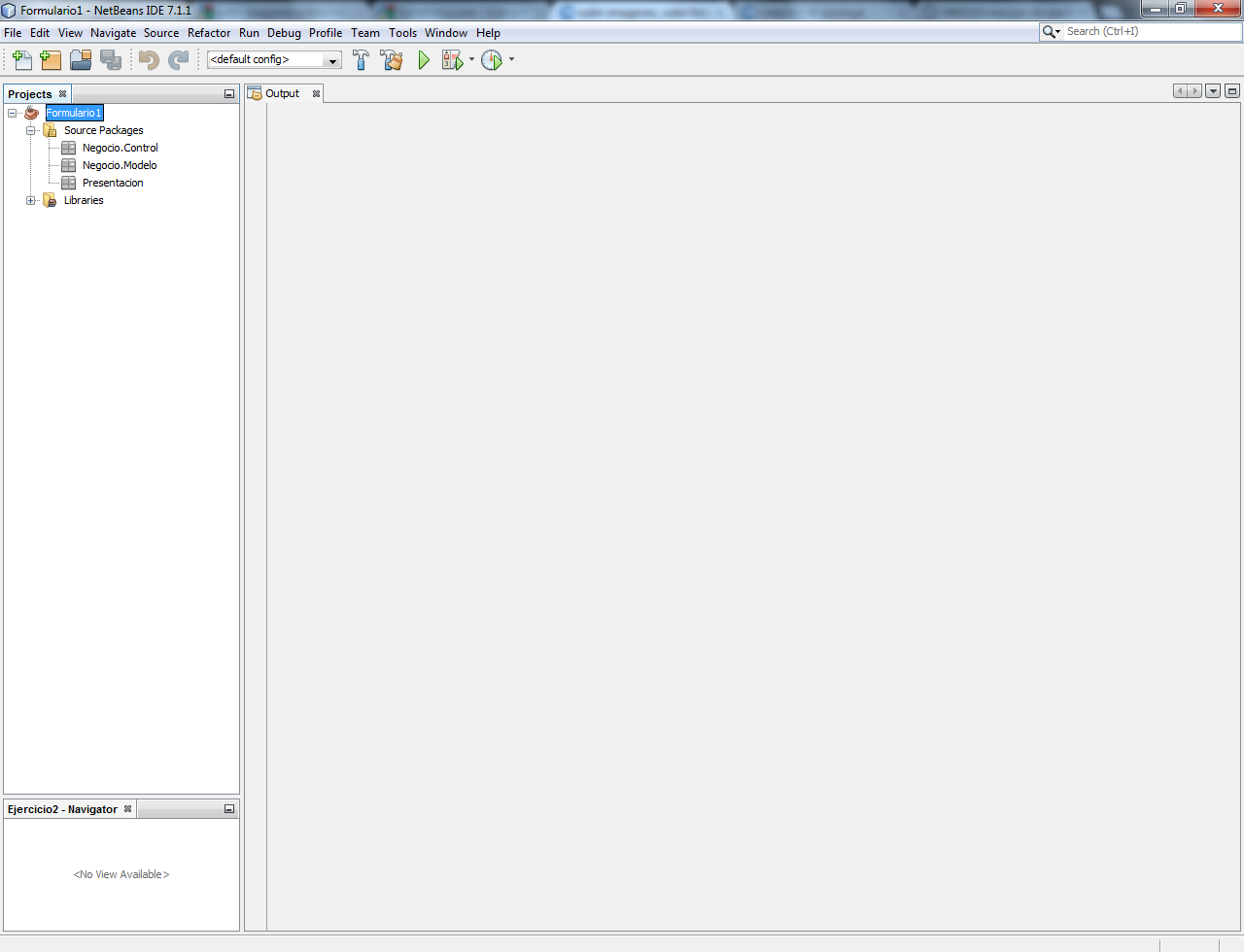
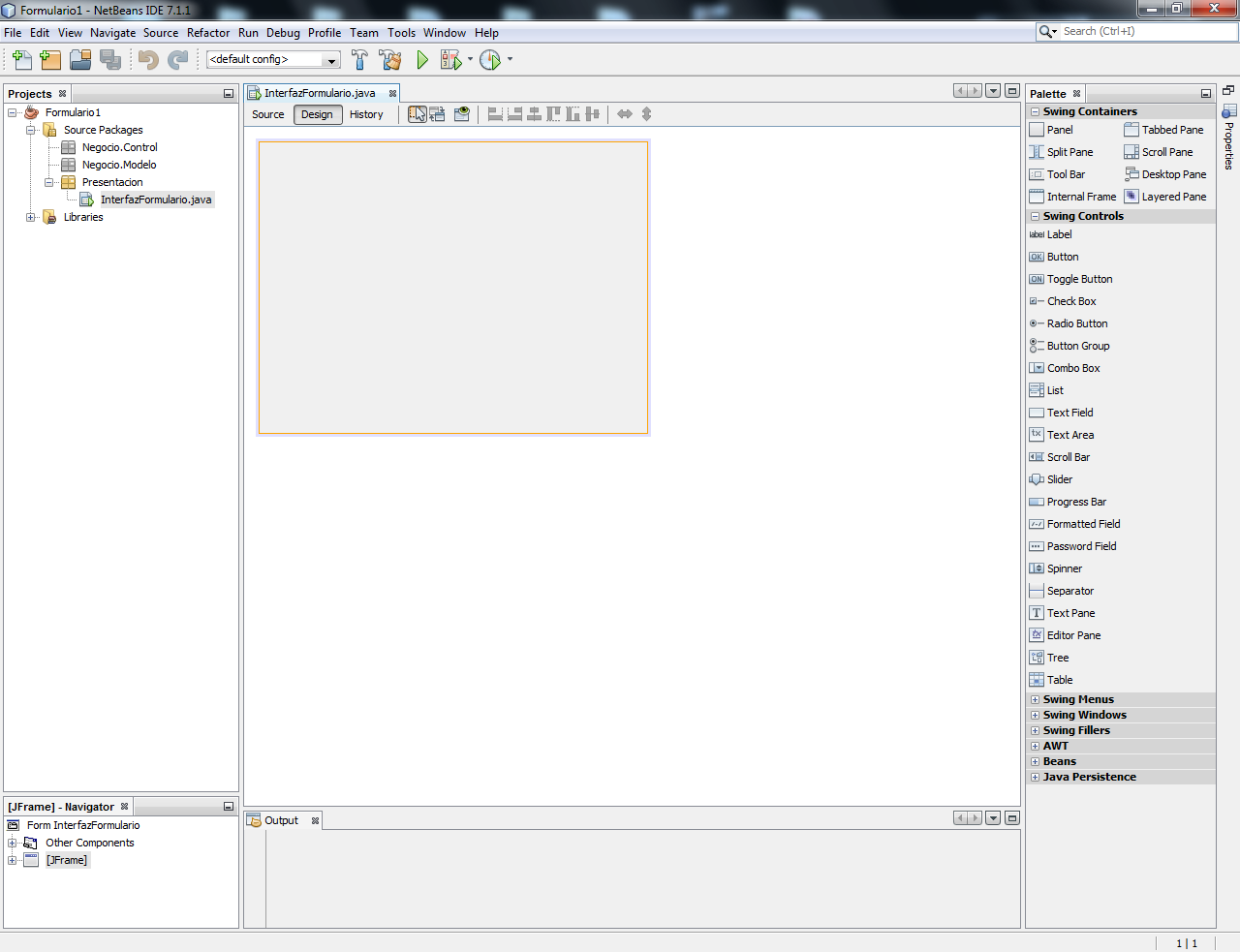
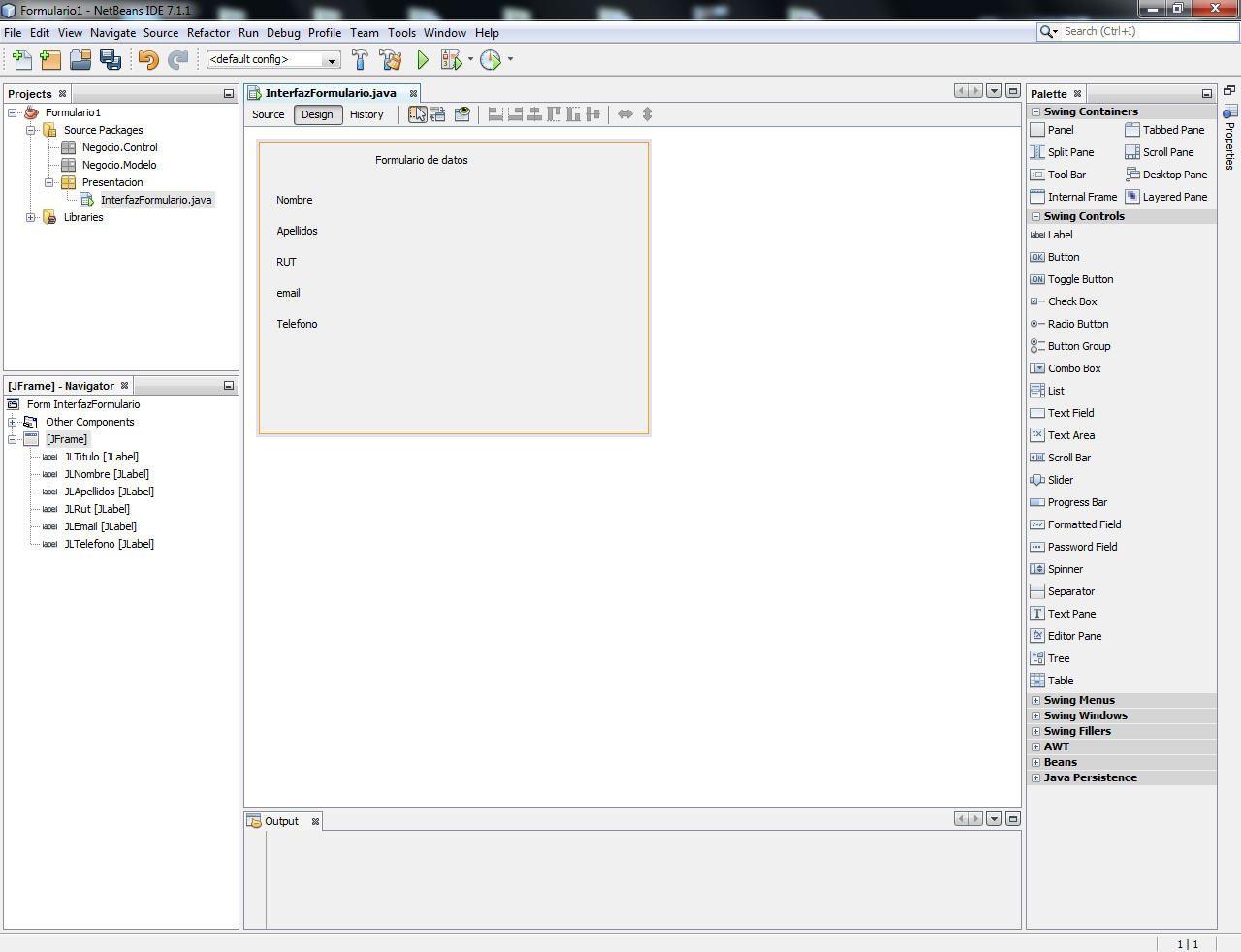
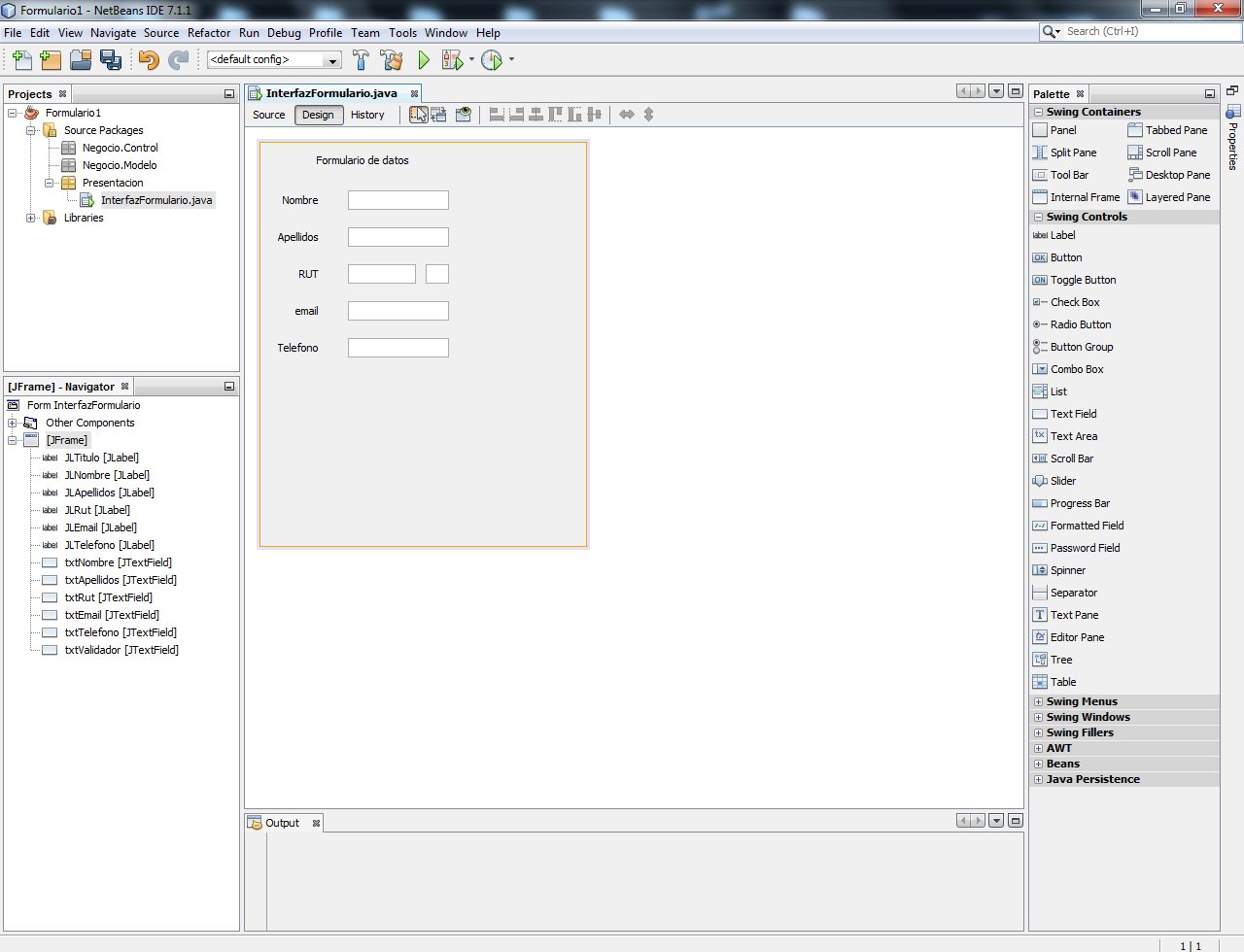
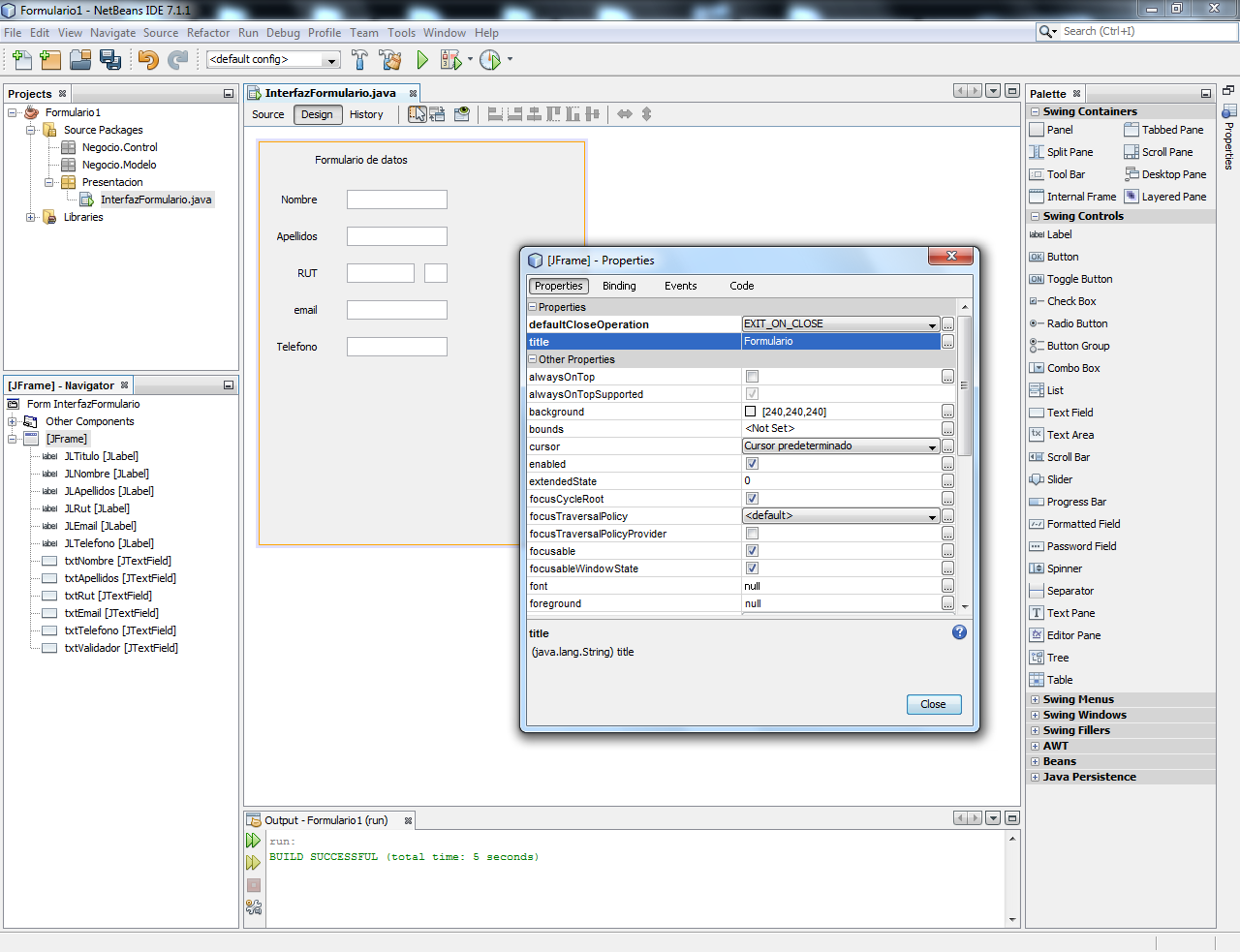
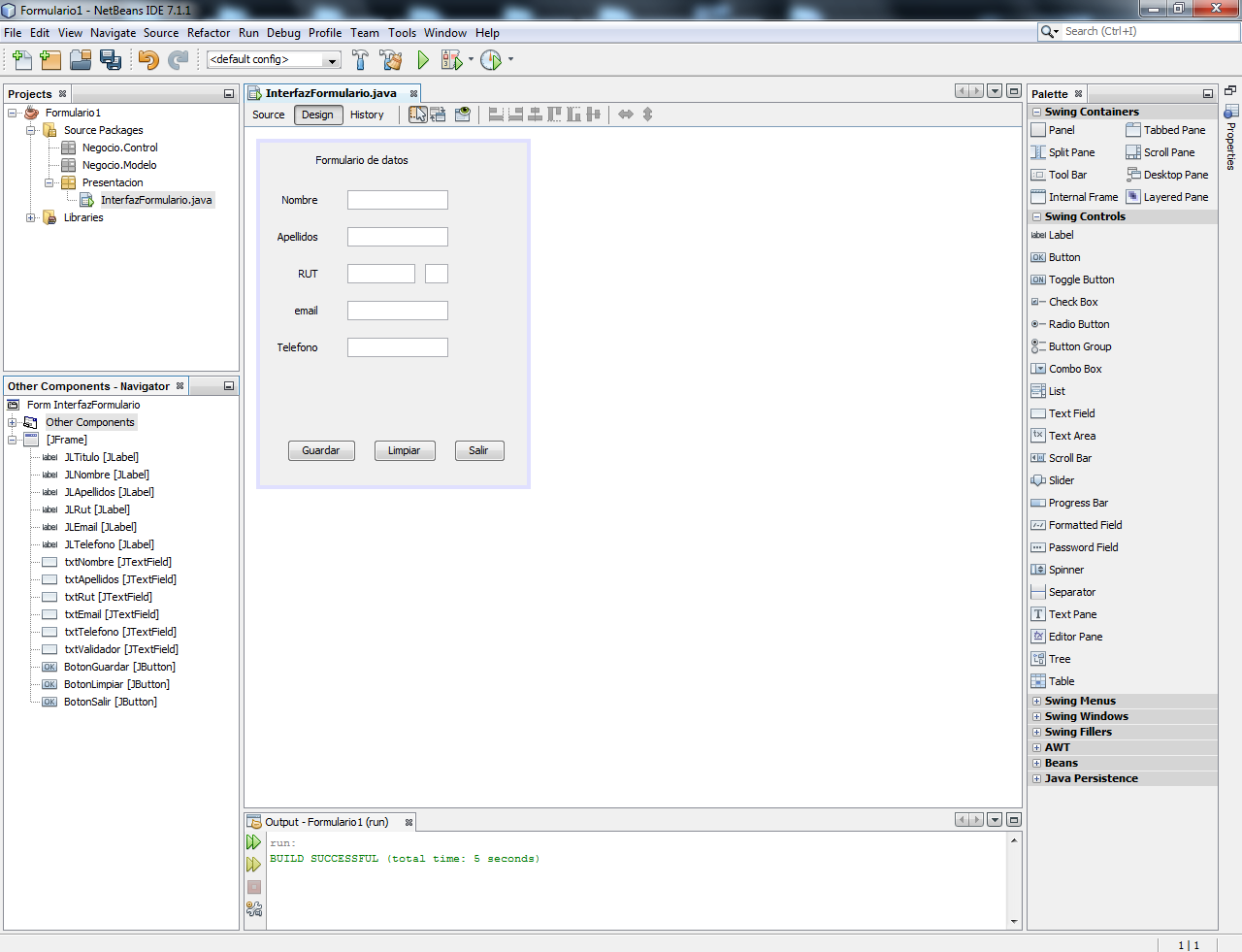
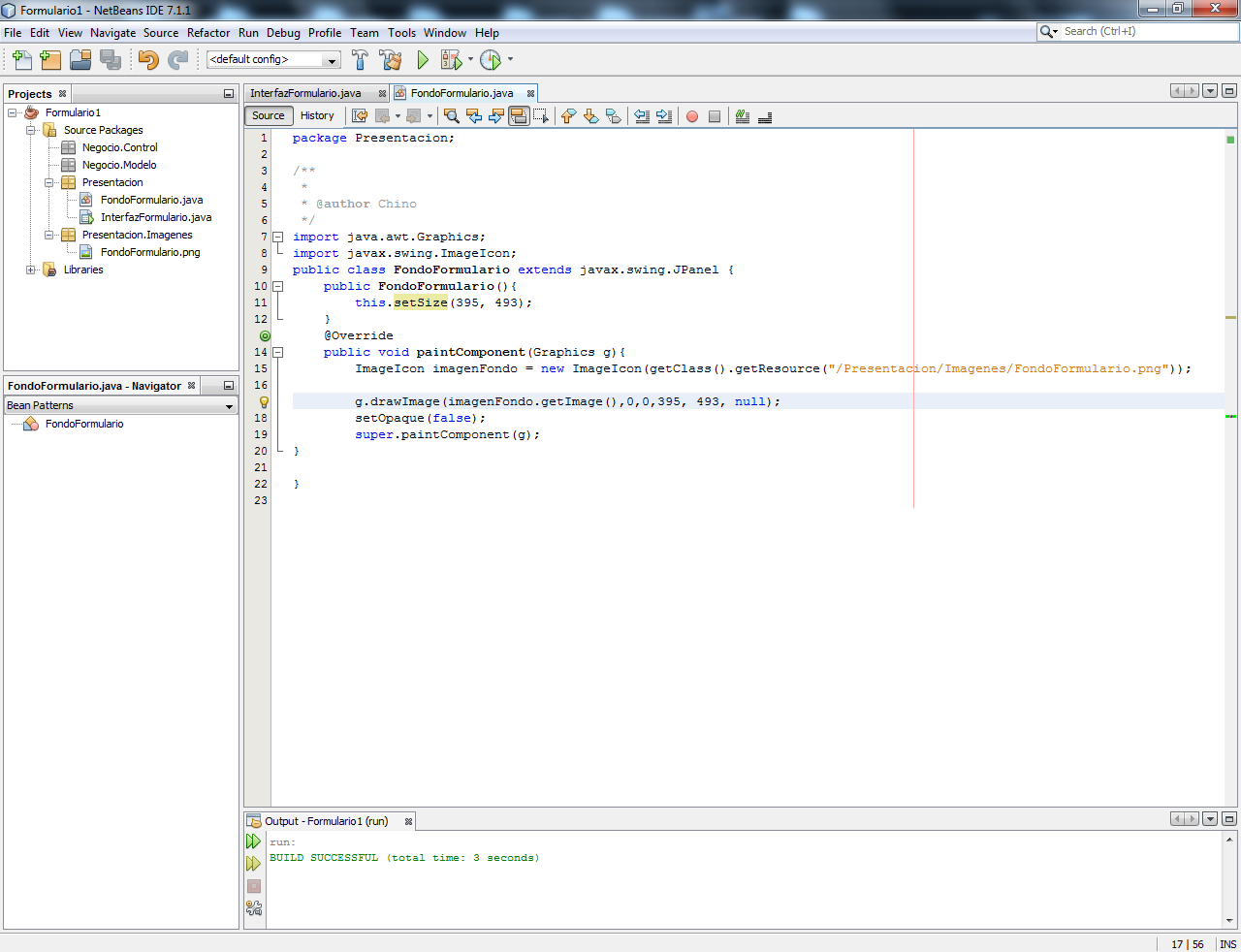
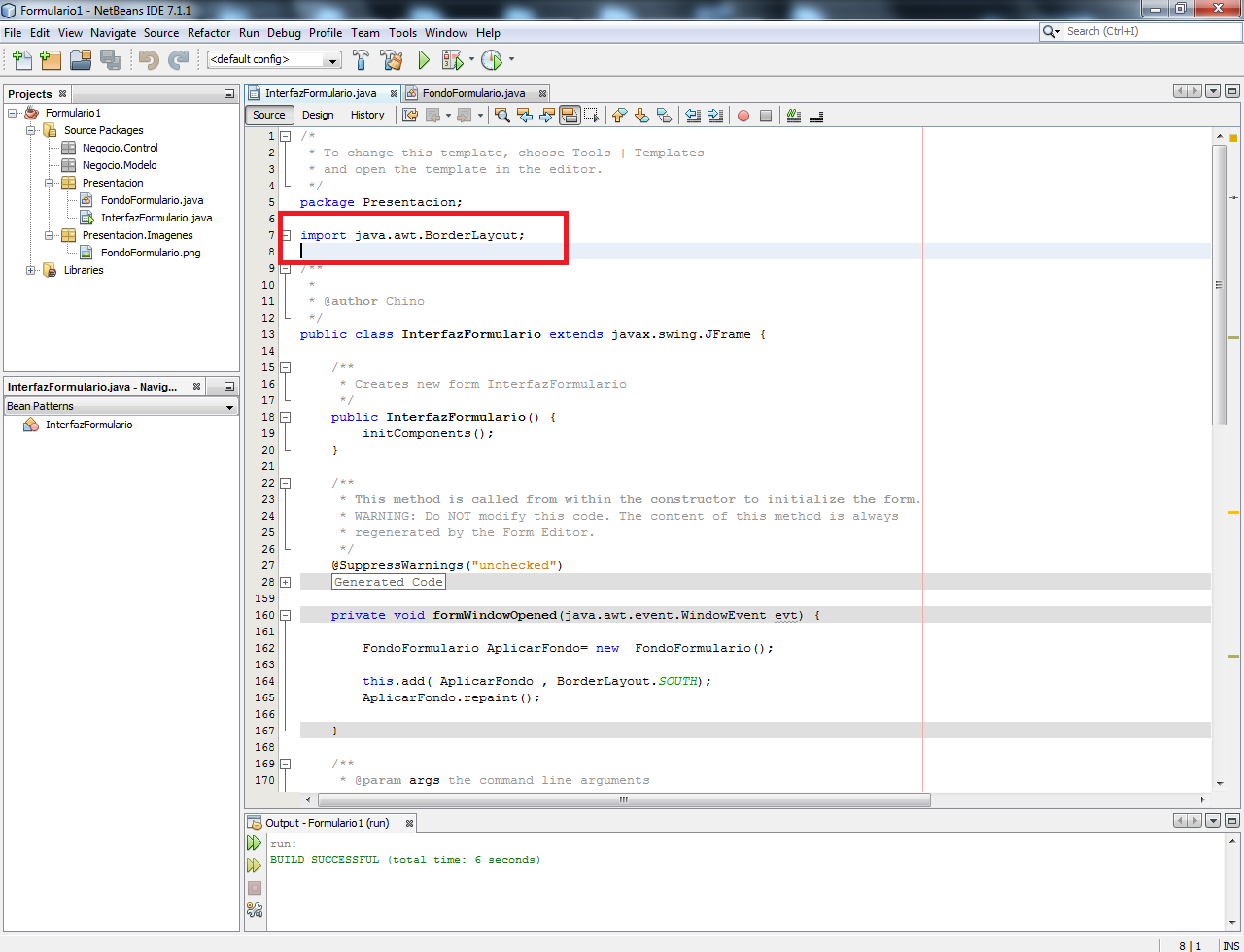
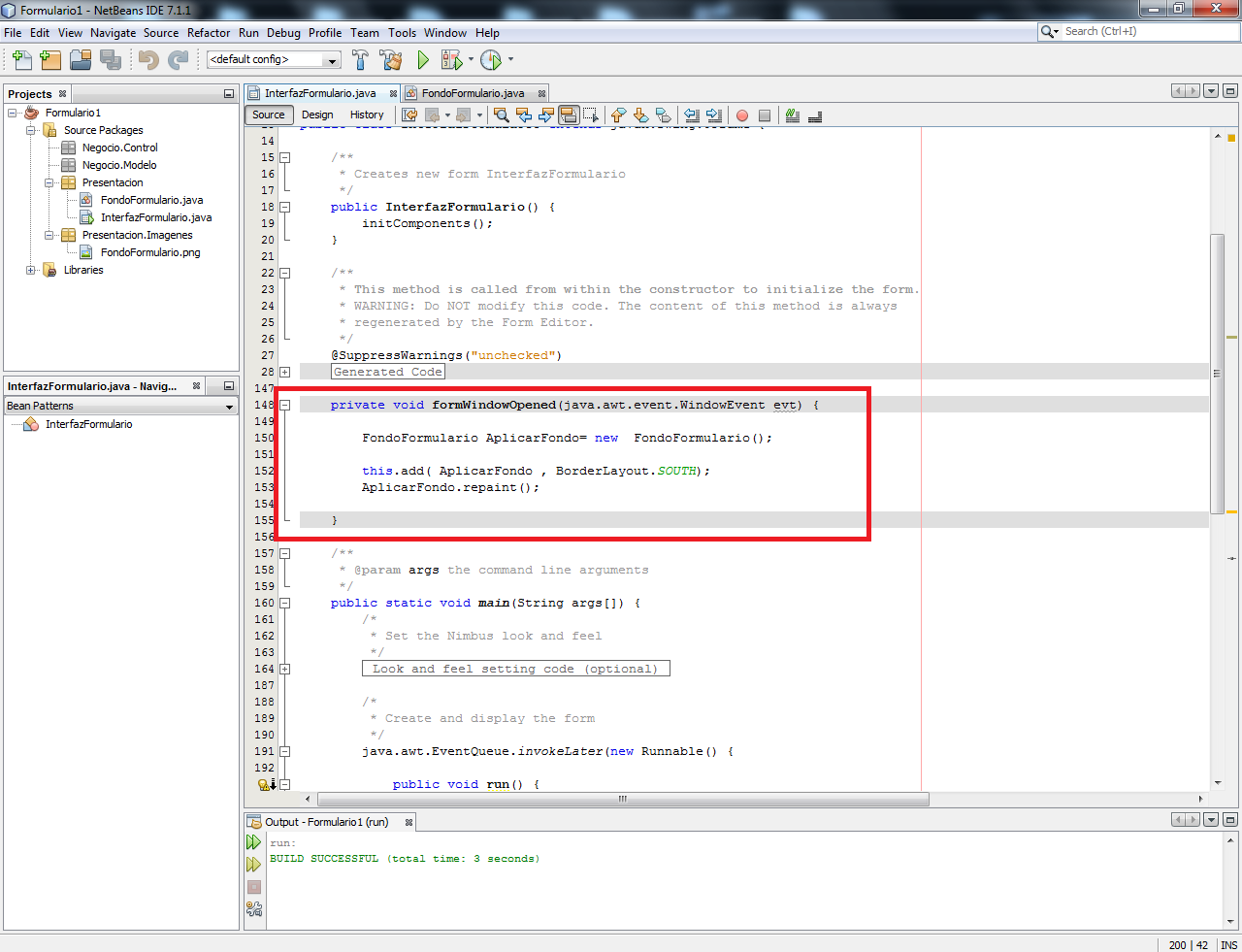
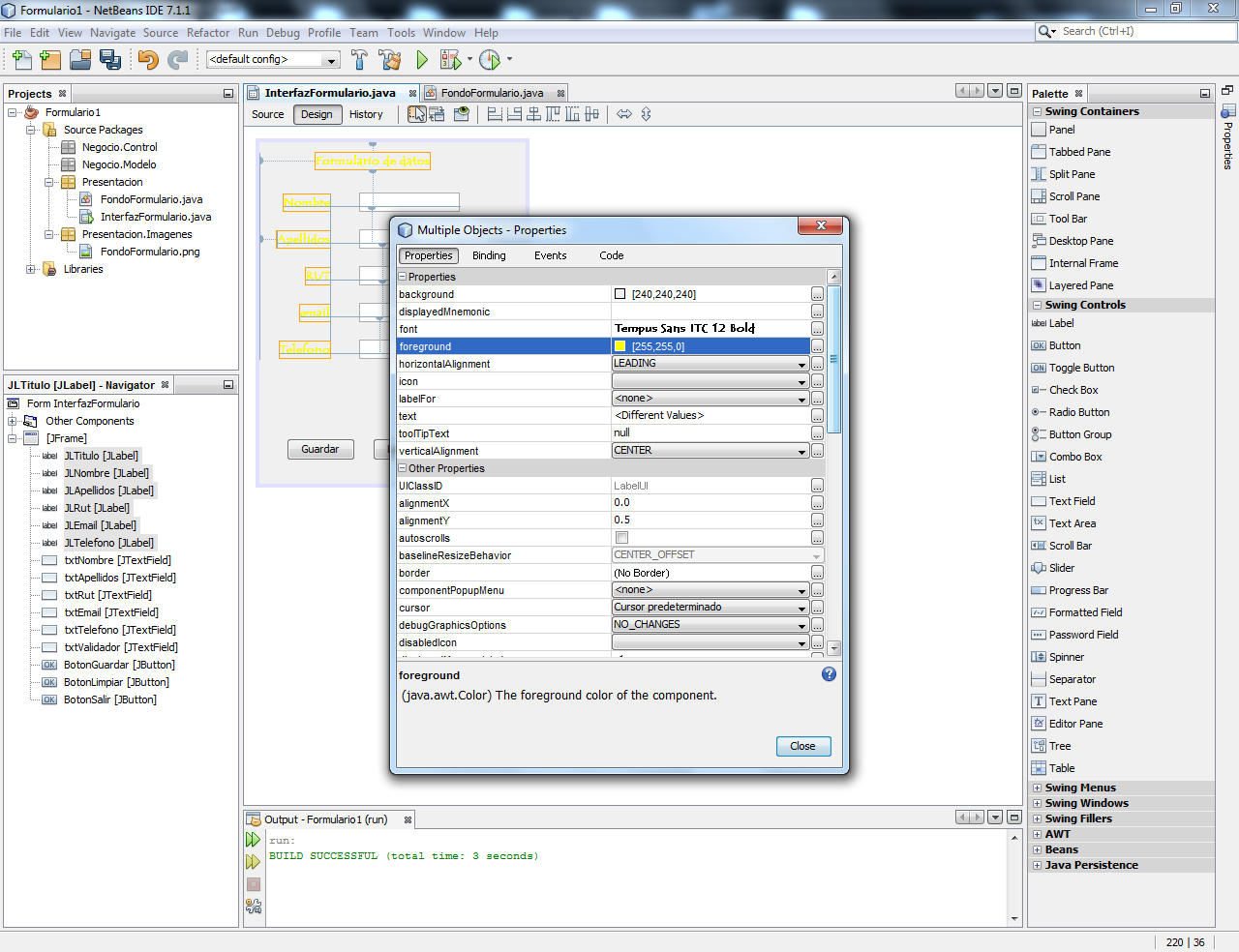
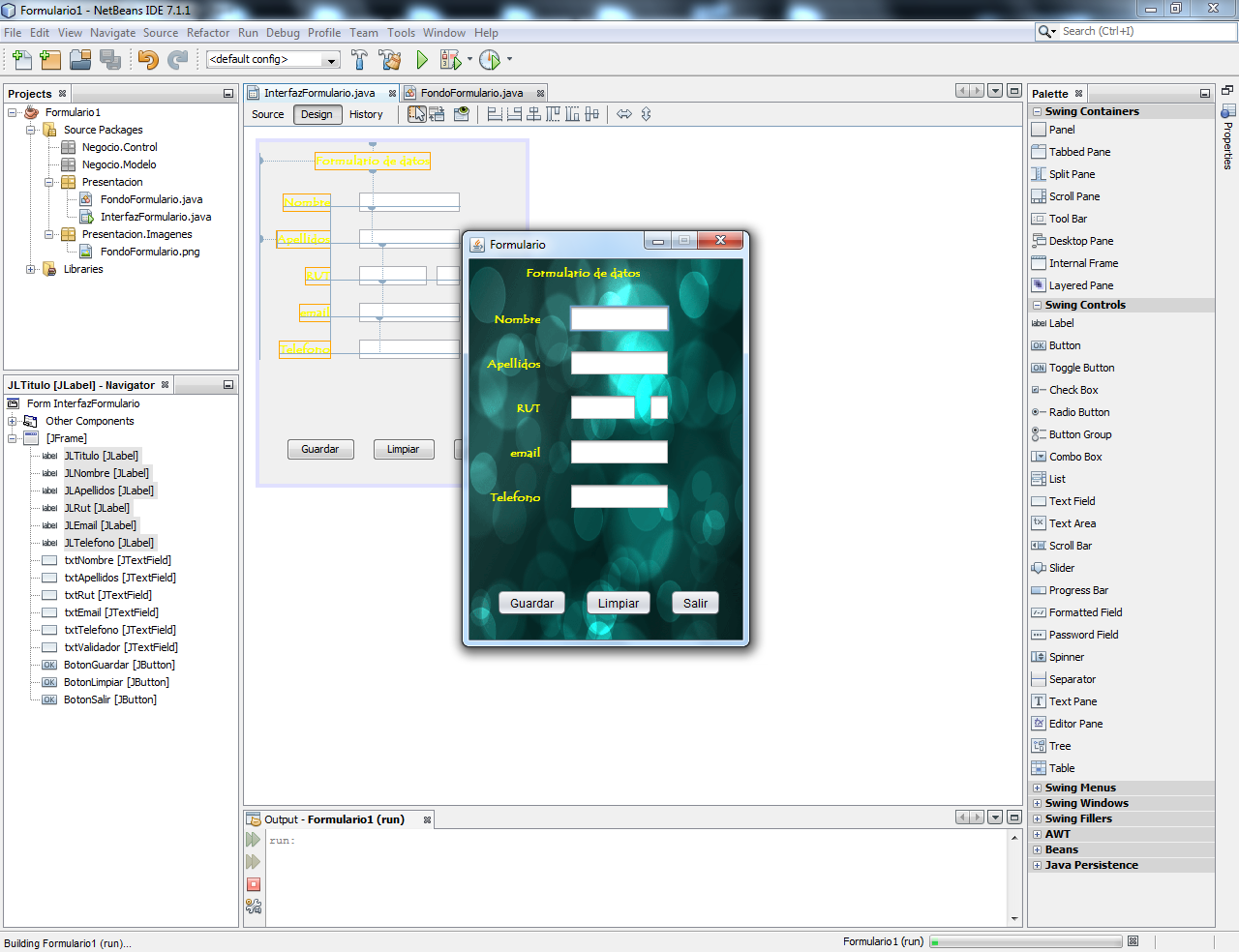
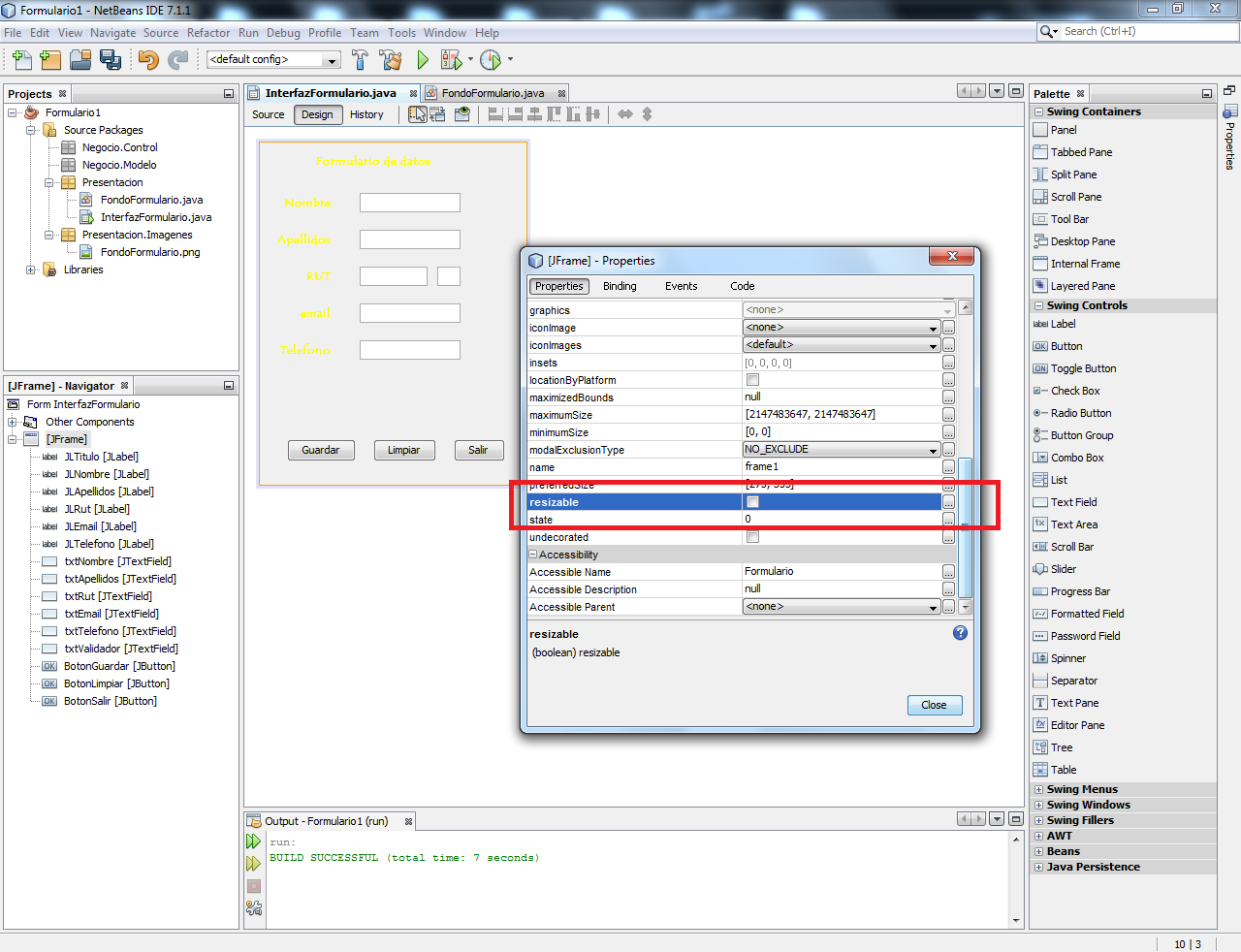
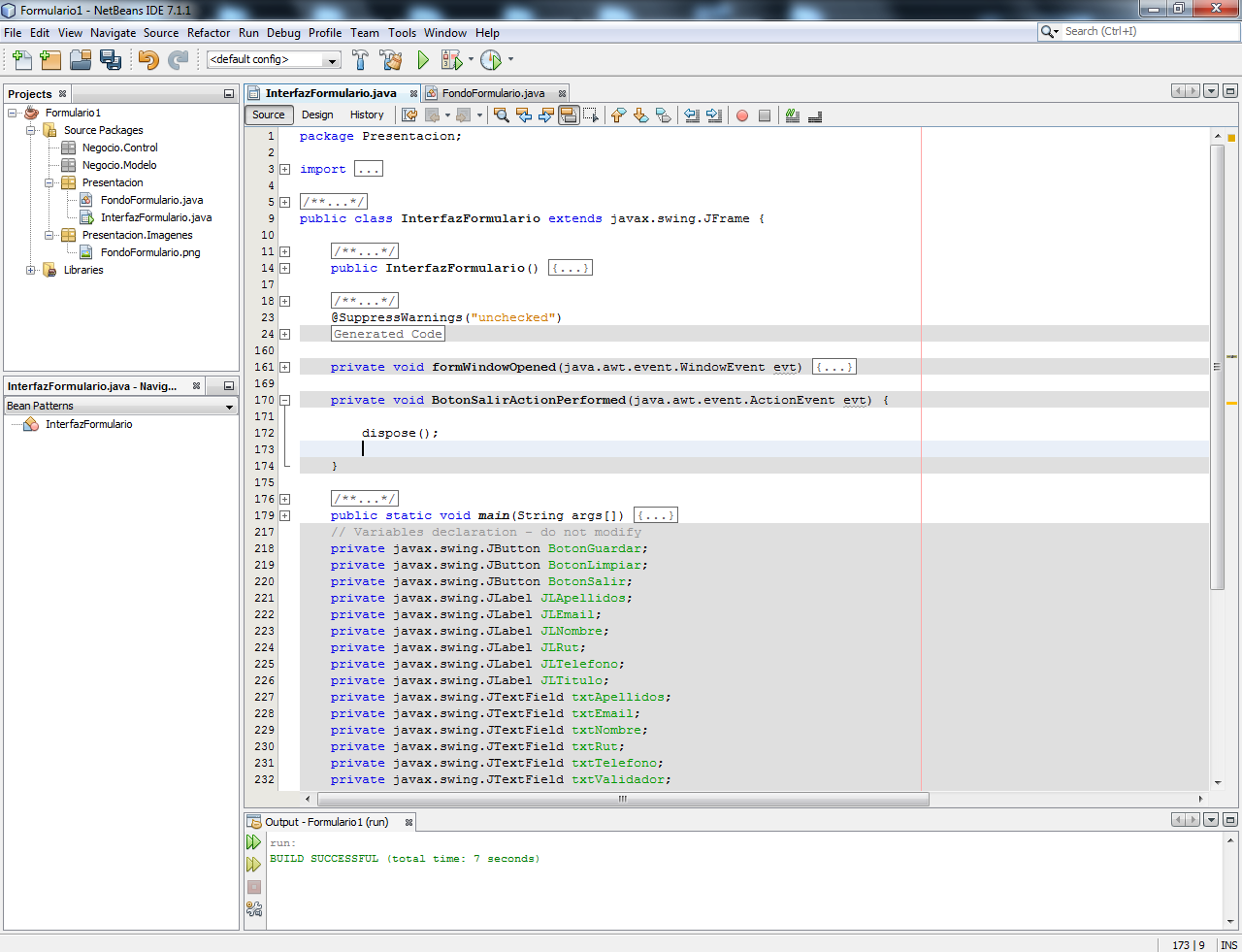
JFrame ventana = new JFrame("Ventana");   
ventana.setSize(800,600);   
ventana.setResizable(false);   
ventana.setVisible(true);

**(Tutorial) Crear un formulario de datos en NetBeans java**

**Objetivos:   
Crear un formulario que pida los datos que el ejercicio requiera, hacer campos obligatorios, agregar imagenes al formulario, entre otras...**  
  
  
lo aré lo mas explicativo posible si ya lo saben hacer sáltense pasos no vomiten insultos...   
  
  
N°1.- Abrir NetBeans y crear un "New Project"--"java"--"Java Class Library".   
(Dar el nombre que uno quiera no influye en nuestro objetivo).   
  
  
  
Para efectos del ejemplo lo llamare Formulario1   
  
N°2.- En nuestro Source Packages, daremos click secundario, y crearemos un package llamado Presentacion,otro llamado Negocio.Modelo y otro llamado Negocio.Control   
(Esto lo hacemos para tener un código mas limpio)   
Nos quedara algo así...   
  
  
  
Trabajaremos en Presentacion primeramente.   
  
Presentacion   
  
N°3.- En nuestro package Presentacion le damos click secundario, seleccionamos "new" y le damos a "JFrame Form". (En caso de no estar abajo sale other... buscan "Swing GUI Forms" y dentro de el tiene que aparecer "JFrame Form" ) .   
  
Lo creare con el nombre de "InterfazFormulario"..   
  
  
  
En este ejemplo pediremos los siguientes datos (Nombre, apellidos, RUT, email, telefono)   
seran obligatorios todos los datos (El rut es para los chilenos asique su codigo lo enseñare al final de este pequeño tutorial).   
  
para este fin tendremos estos jlabel:   
Nombre, Apellidos, Rut, Email, Telefono, y que sus Nombres de variables seran "JLNombre" en el caso del nombre...   
(en caso de tener titulo crear un jlabel con el titulo).   
  
Ponerle nombre visible por el usuario es click secundario y "Edit Text"   
Ponerle nombre a la variable es click secundario y "Change Variable Name"   
  
Nos quedara algo asi:   
  
  
  
N°4.-Crearemos los JTextField:   
uno para cada Dato, excepto para el rut que necesita 2, con el siguiente nombre de varible   
por ejemplo: para el Nombre se llamara "txtNombre" para el Apellido se llamara "txtApellidos",   
(RUT; txtRut .... txtValidador).   
  
  
le damos click secundario "Edit Text" y le borramos el contenido sin ningun espacio luego agrandamos el textfield en caso de achicarse este...   
  
  
Nos quedara algo así...   
  
  
  
  
N°5.- Le damos un titulo al JFrame eso se hace dandolo click secundario encima del frame en cualquier lugar que no sea ni un jlabel o un JtextField y damos en "Propiedades", luego en donde dice "title" escribimos "Formulario"   
  
  
  
N°6.- Ahora crearemos botones "Button". crearemos 3 uno para "Guardar" uno para "Limpiar" y uno para "Salir" en los cuales al igual que en los jlabel y textfield vamos a ponerle nombre de variable...   
Guardar = BotonGuardar; Limpiar = BotonLimpiar; Salir = BotonSalir.   
  
  
  
N°7.- Agregaremos fondo a un Formulario, esto lo aremos con una clase y una imagen...   
Crearemos una clase llamada "FondoFormulario" (Java Class, en Presentacion) y a su vez crearemos un nuevo package llamado "Presentacion.Imagenes" (donde pondremos todas las imagenes).   
  
en FondoFormulario.java pondremos el siguiente codigo:   
  
//Inicio del código 

***package Presentacion;   
  
/\*\*   
\*   
\* @ author Chino   
\*/   
import java.awt.Graphics;   
import javax.swing.ImageIcon;   
public class FondoFormulario extends javax.swing.JPanel {   
public FondoFormulario(){   
this.setSize(395, 493);   
}   
@ Override   
public void paintComponent(Graphics g){   
ImageIcon imagenFondo = new ImageIcon(getClass().getResource("/Presentacion/Imagenes/FondoFormulario.png" ) ) ;   
  
g.drawImage(imagenFondo.getImage(),0,0,395, 493, null);   
setOpaque(false);   
super.paintComponent(g);   
}   
  
}***

//Fin del código   
  
Nos quedara algo así...   
  
  
  
hasta el momento no hemos programado nada solamente nos hemos centrado en lo gráfico y aun no tiene fondo nuestro formulario, así que ahora empezaremos con un poco para programar que se ponga la imagen de fondo....   
  
N°8.-   
  
Le damos Click secundario sobre el JForm principal y vamos a "Events" -- "Window" -- "windowOpened"   
  
y ahí instanciamos la clase que posee el fondo y la aplicamos de la siguiente manera..   
  
(deberan importar la libreria "import java.awt.BorderLayout ; " eso se debe escribir abajo del nombre del package )   
  
  
  
  
  
N°9.- Un detalle importante tendremos que cambiar de color los JLABEL dado que el fondo que escogí es oscuro por lo tanto las dejare amarillas y para efectos del ejemplo también sirve que vean como se hace   
Click secundario sobre el JLabel a cambiar el color y "Propiedades"   
  
  
  
y ahora al ejecutarlo obtendremos el siguiente resultado...   
  
  
  
N°10.- Pero su tamaño puede ser cambiado por el usuario, por lo que debemos meternos denuevo a las propiedades del JForm principal y cambiar esa virtud...   
  
  
  
Por fin en estos 10 pasos hemos terminado la parte netamente gráfica de este asunto.   
  
Ahora la parte programable de este Proyecto...   
  
N°1.- Empezaremos por el mas facil el BotonSalir que le daremos click secundario iremos a "Events" - "Action" - "ActionPerformed".   
  
y escribiremos ahí "dispose ( ) ; " o "System.exit(0) ;" o "this.dispose ( ) ;" cualquiera nos sirve pero deben elegir una y dado que solo es una ventana YO elijo "dispose ( ) ;"   
  
  
  
N°2.- Ahora Programaremos el BotonLimpiar con el cual dejaremos todos los JTextField nuevamente sin ninguna escritura...   
Esto se hace de la siguiente manera   
ej:   
txtNombre.setText( "" ) ;   
con esto se dice que a la variable txtNombre le llevamos (set) un String vacio ( "" ) y asi con todas las variables..   
  
nos quedaria un codigo así: 

Fondo jpanel

public class CustomPanel extends JPanel{

private URL url = getClass().getResource("/img/java.png");

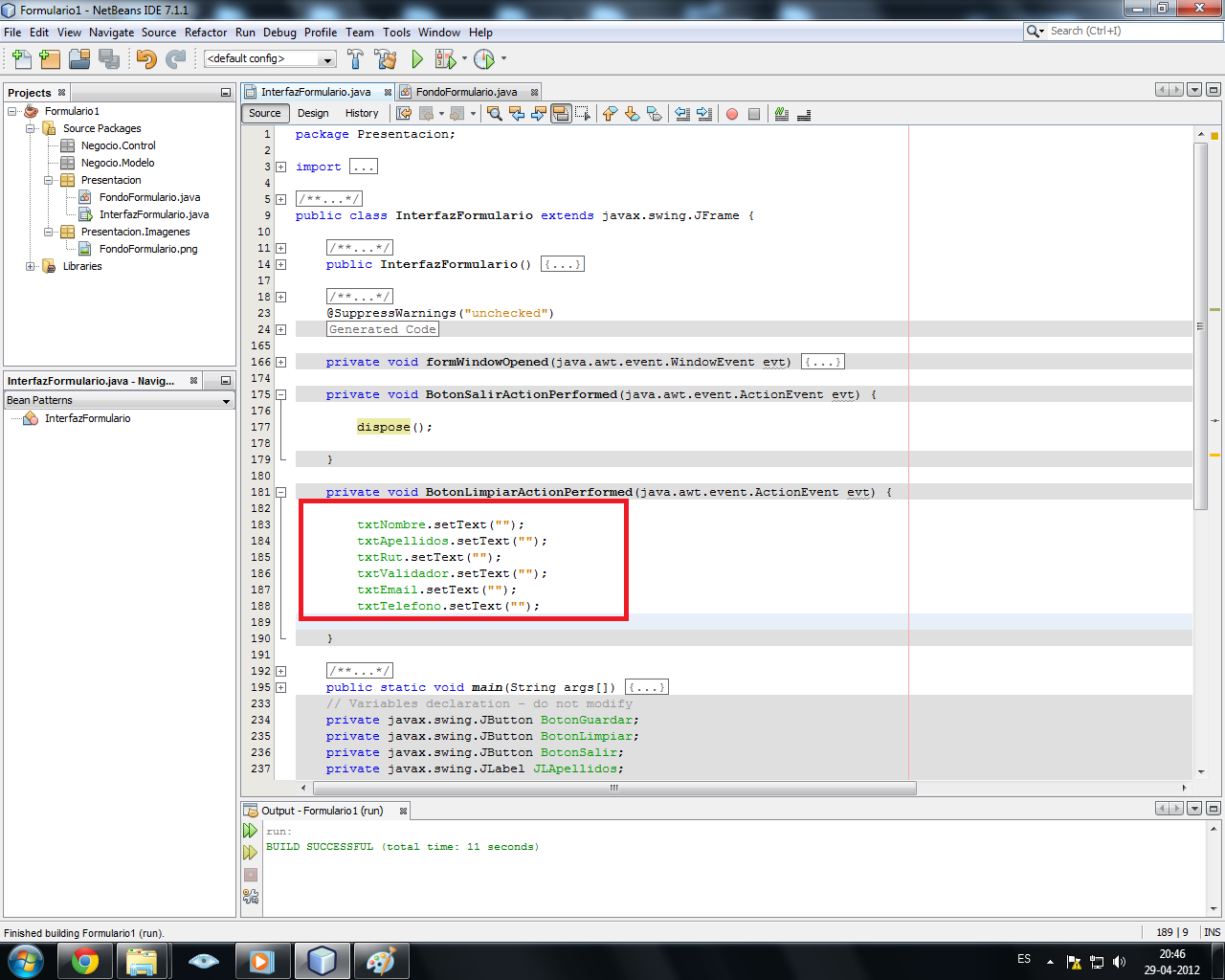
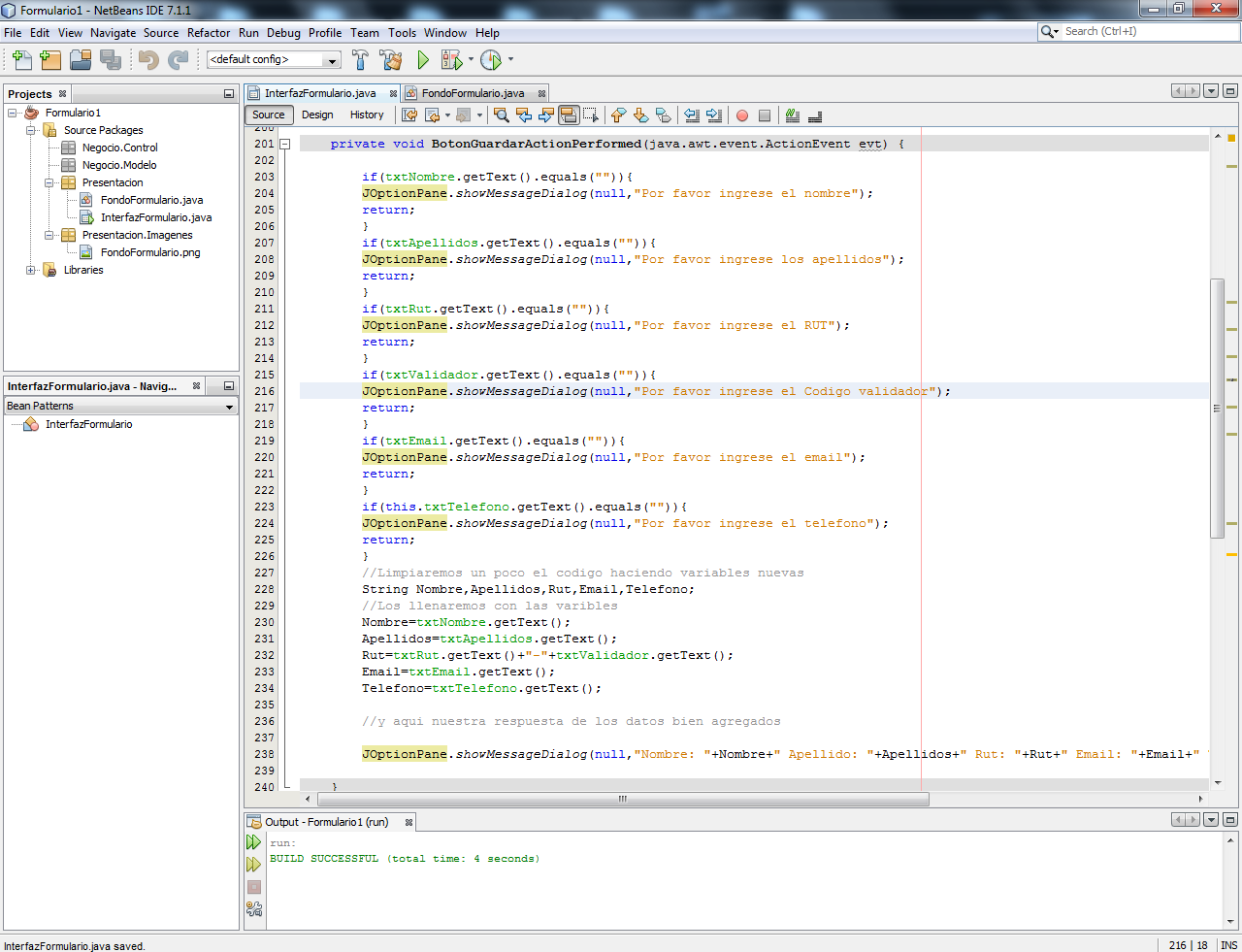
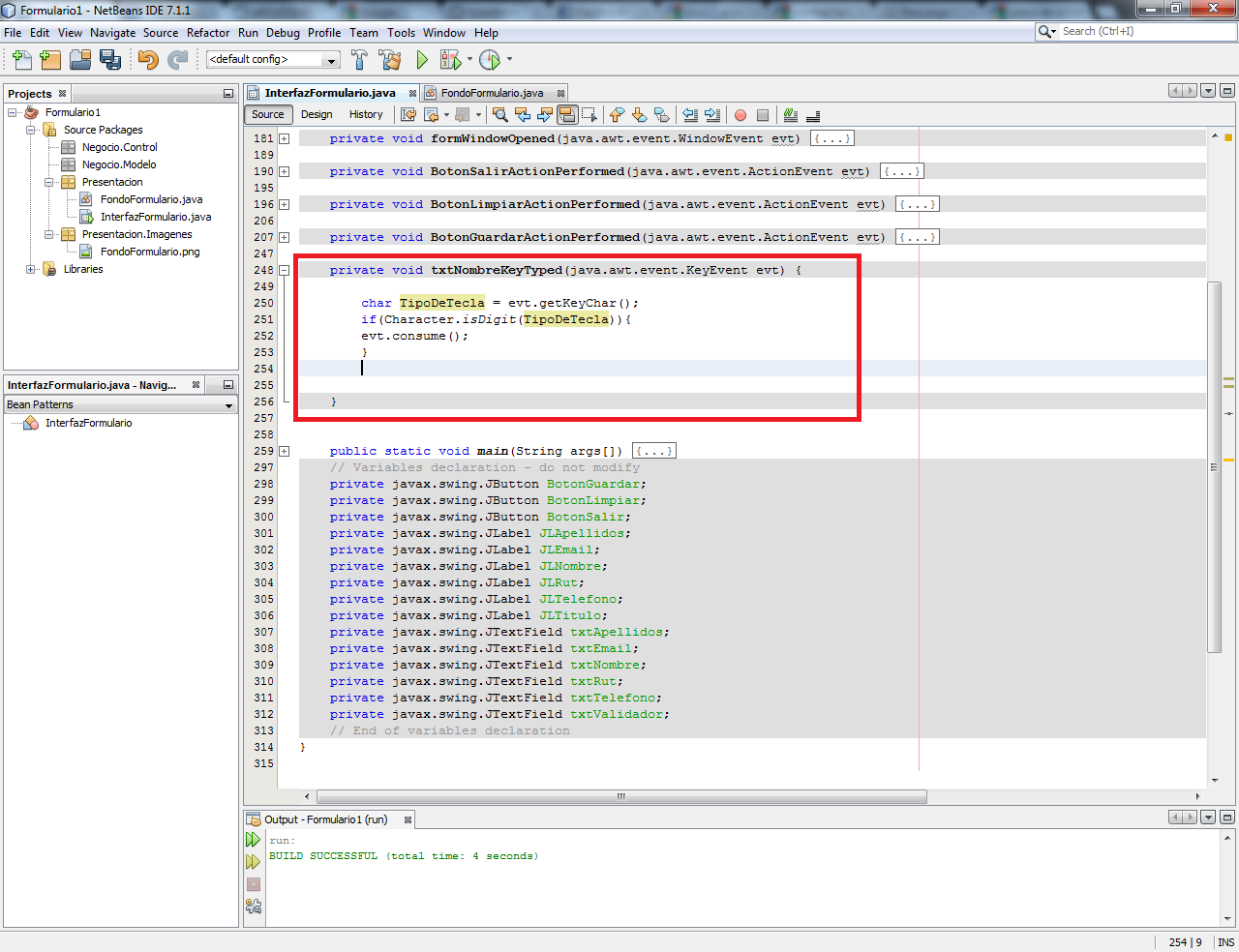
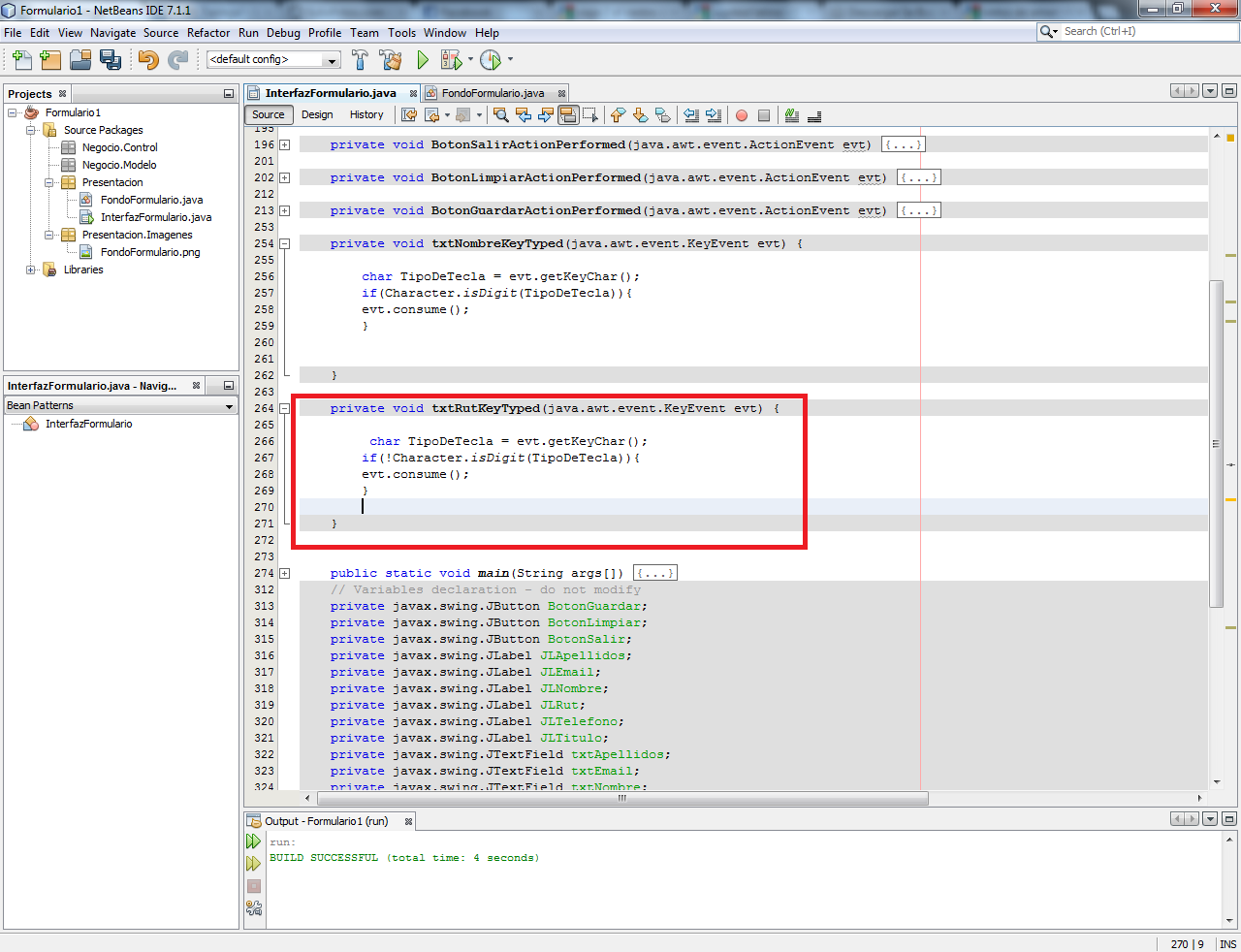
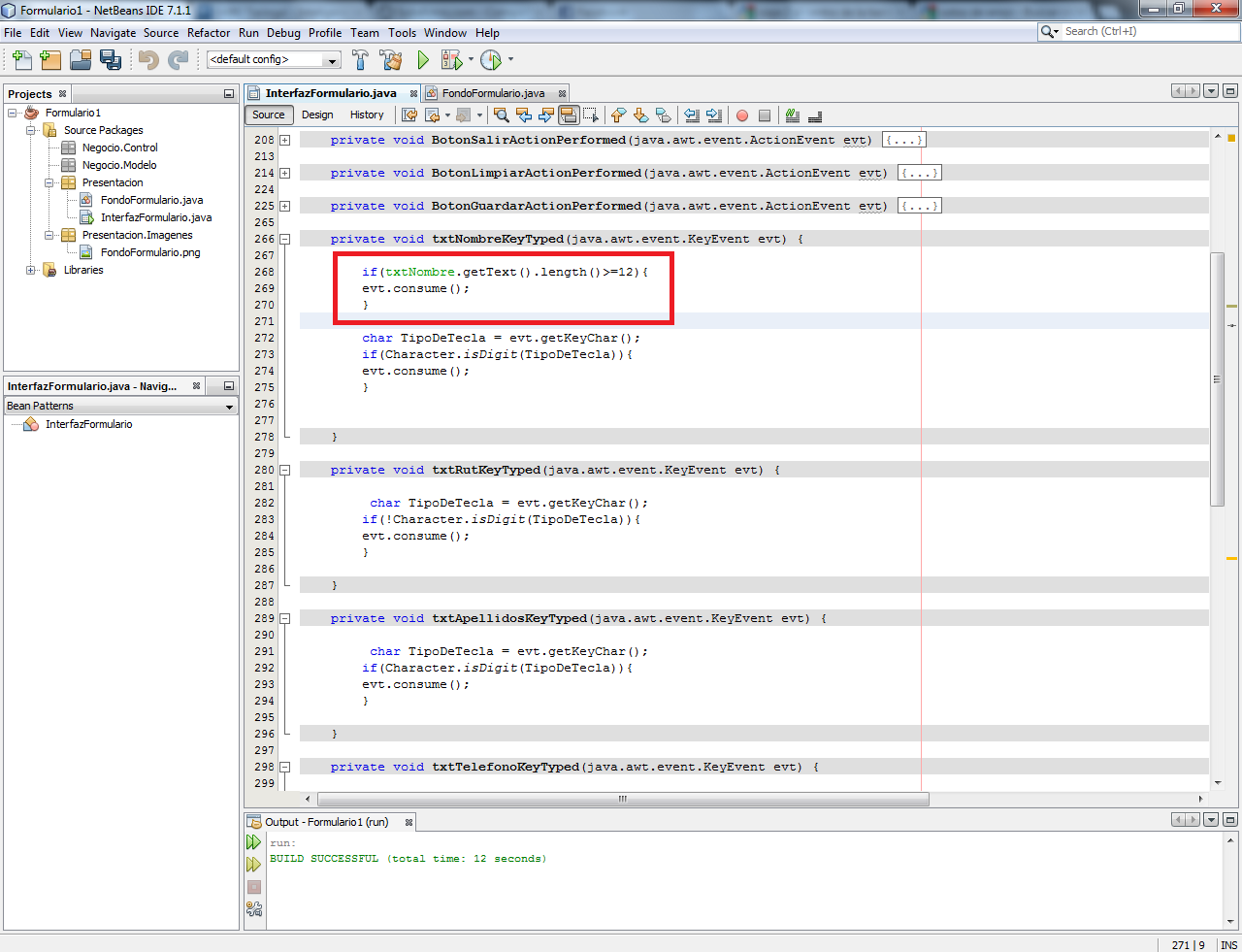
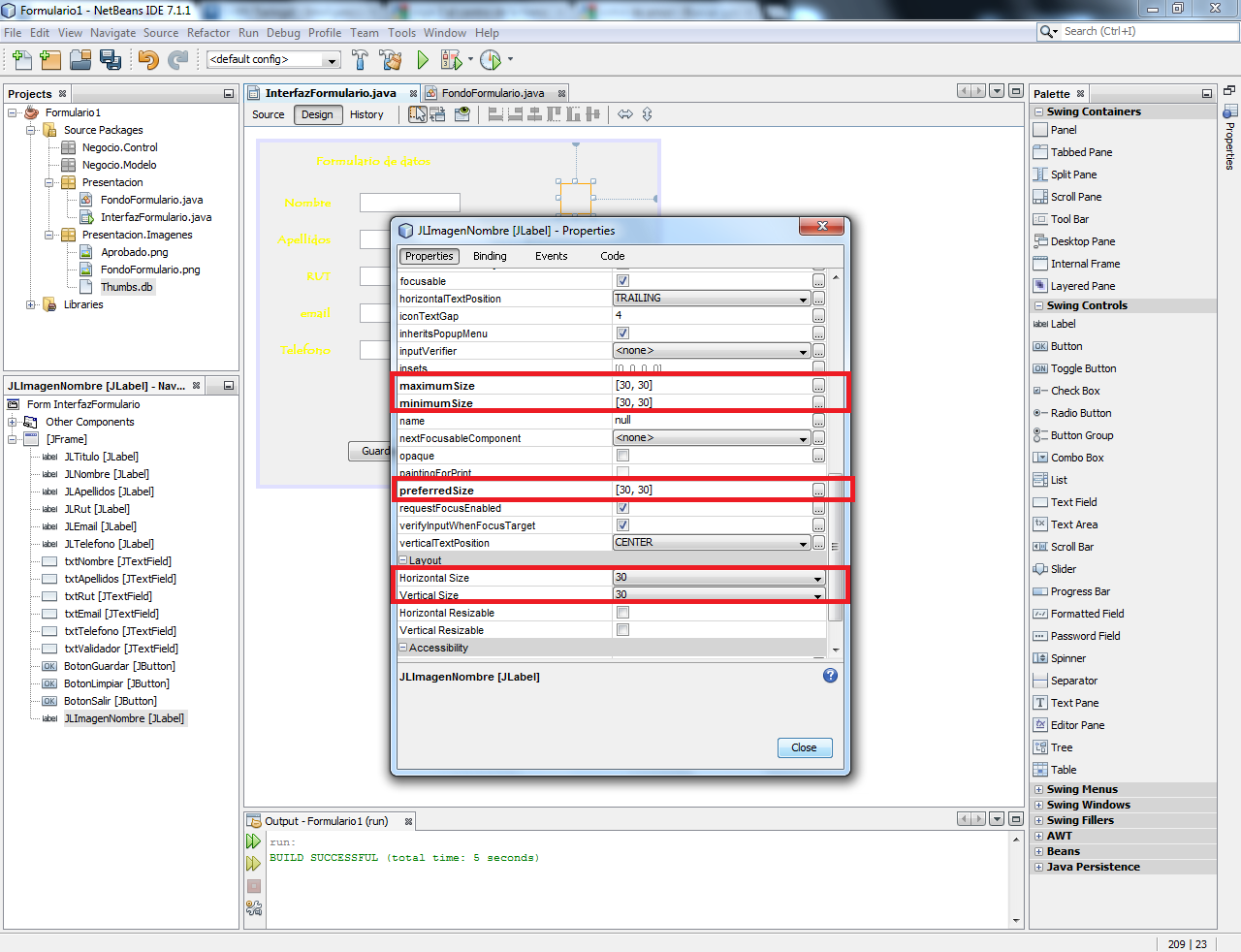
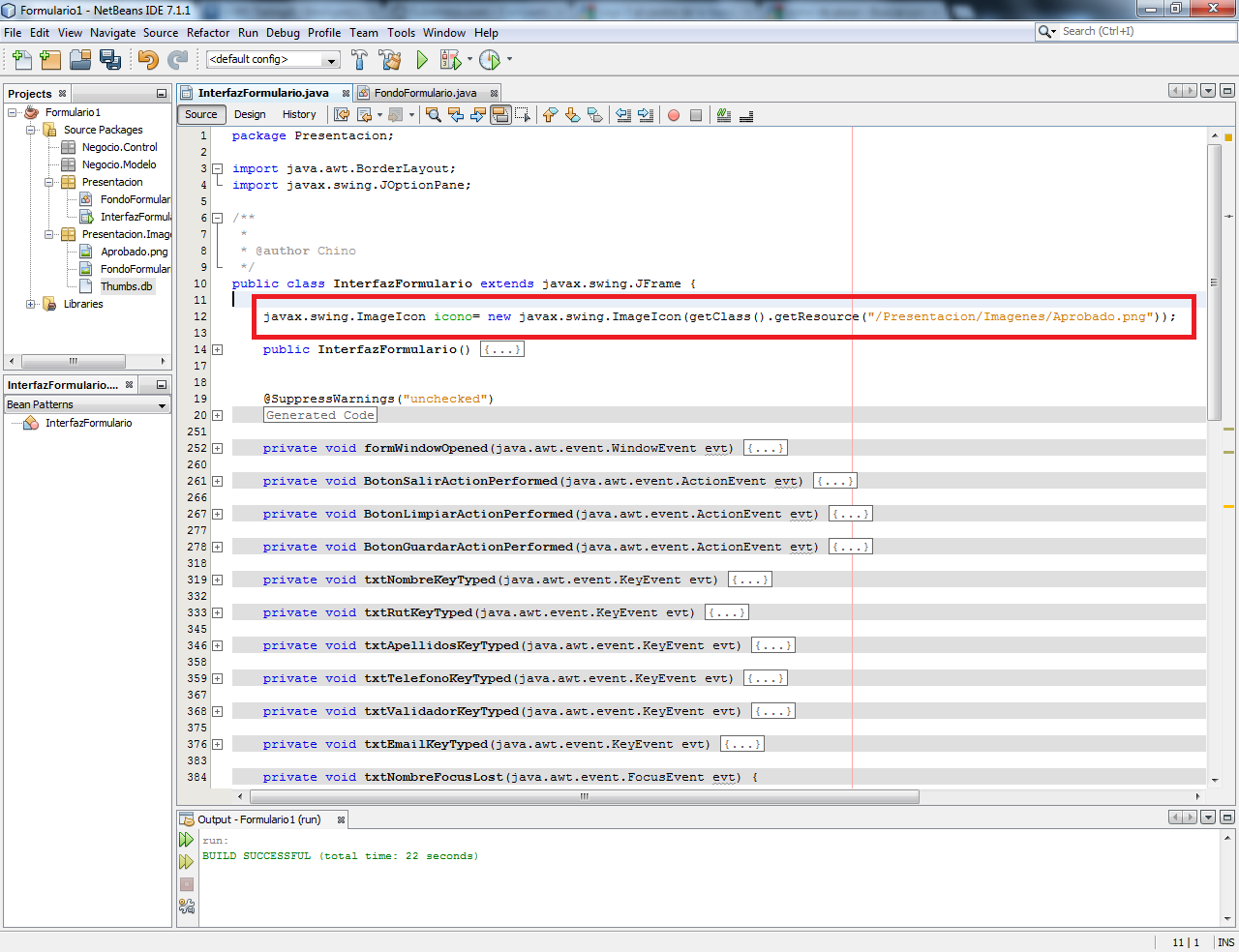
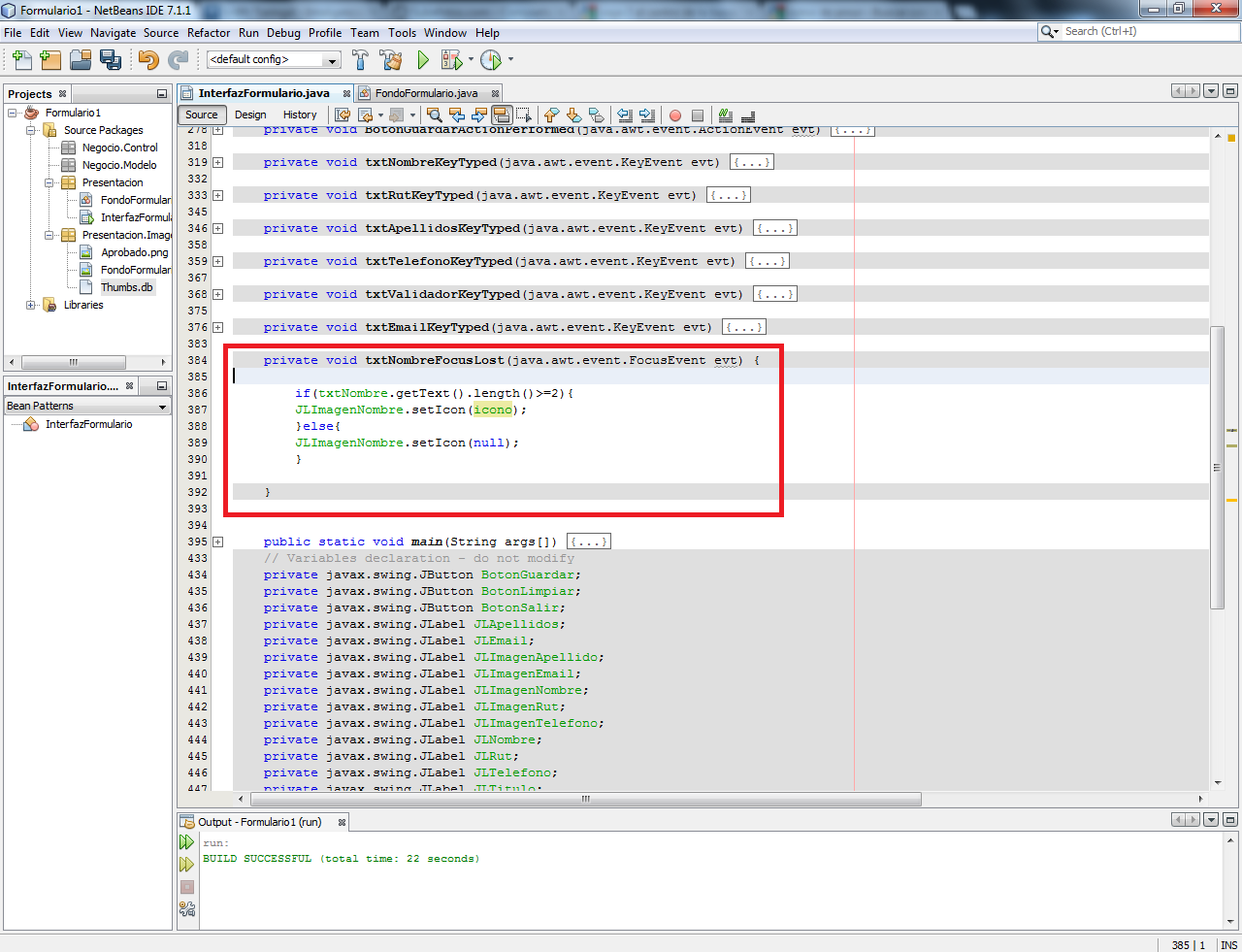
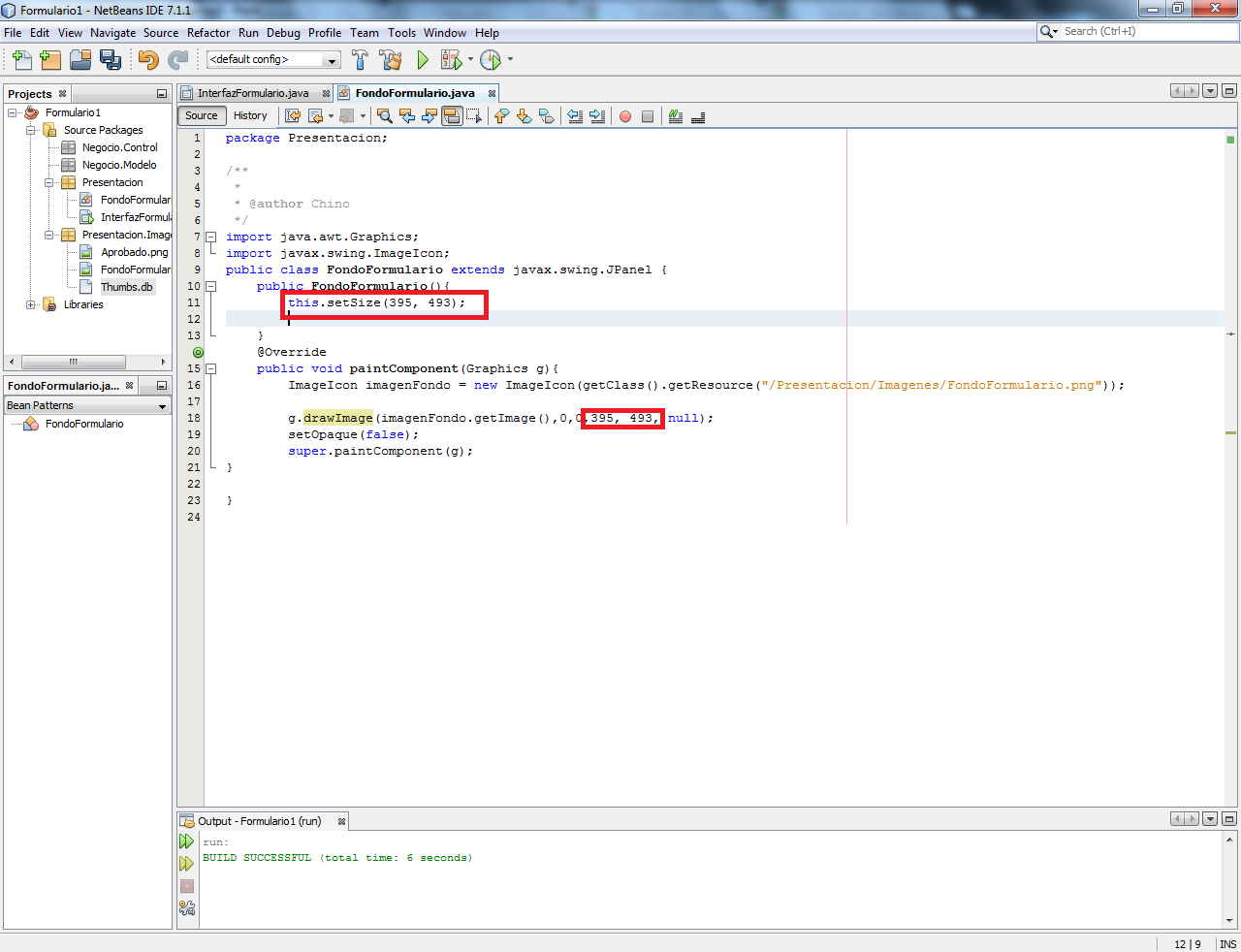
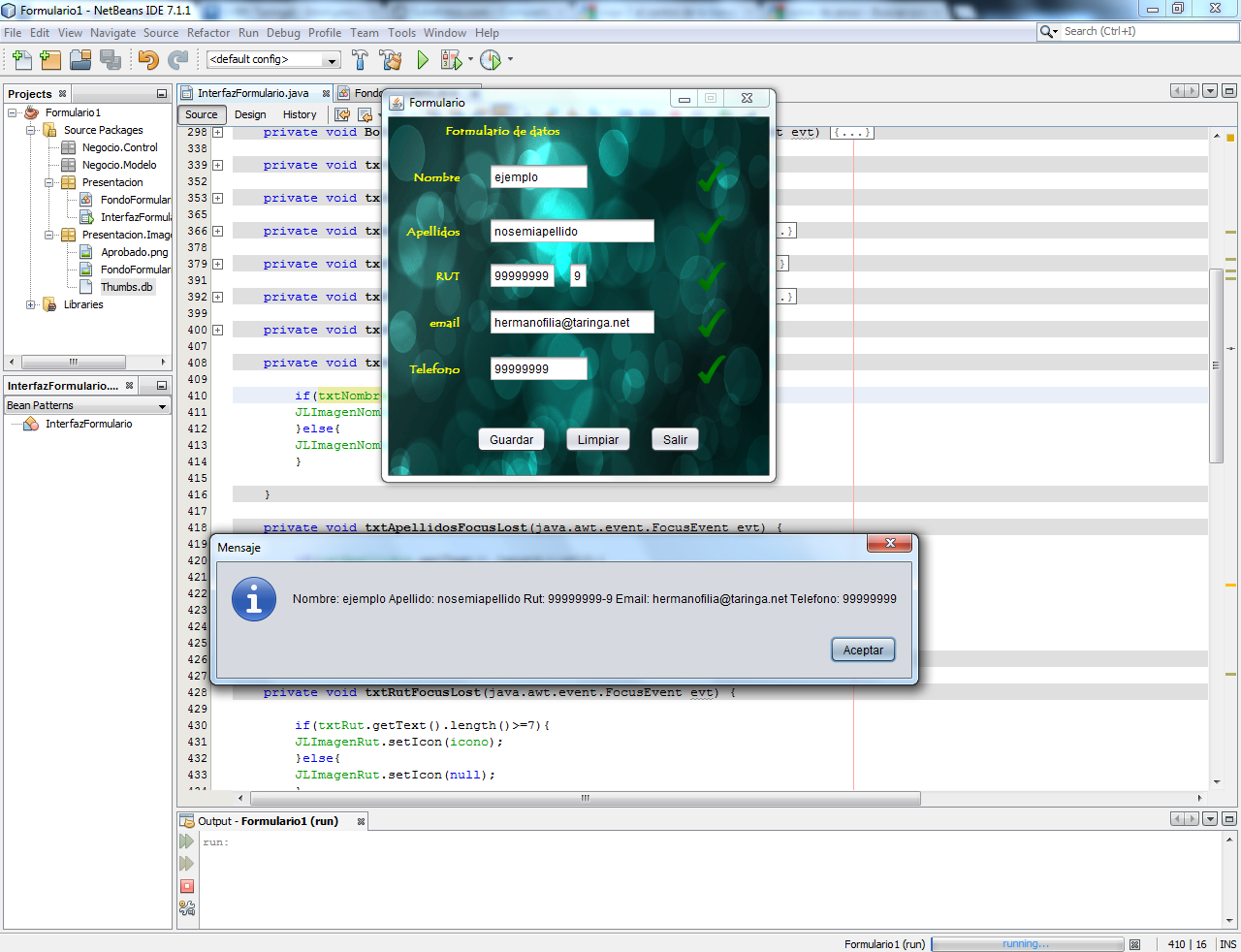
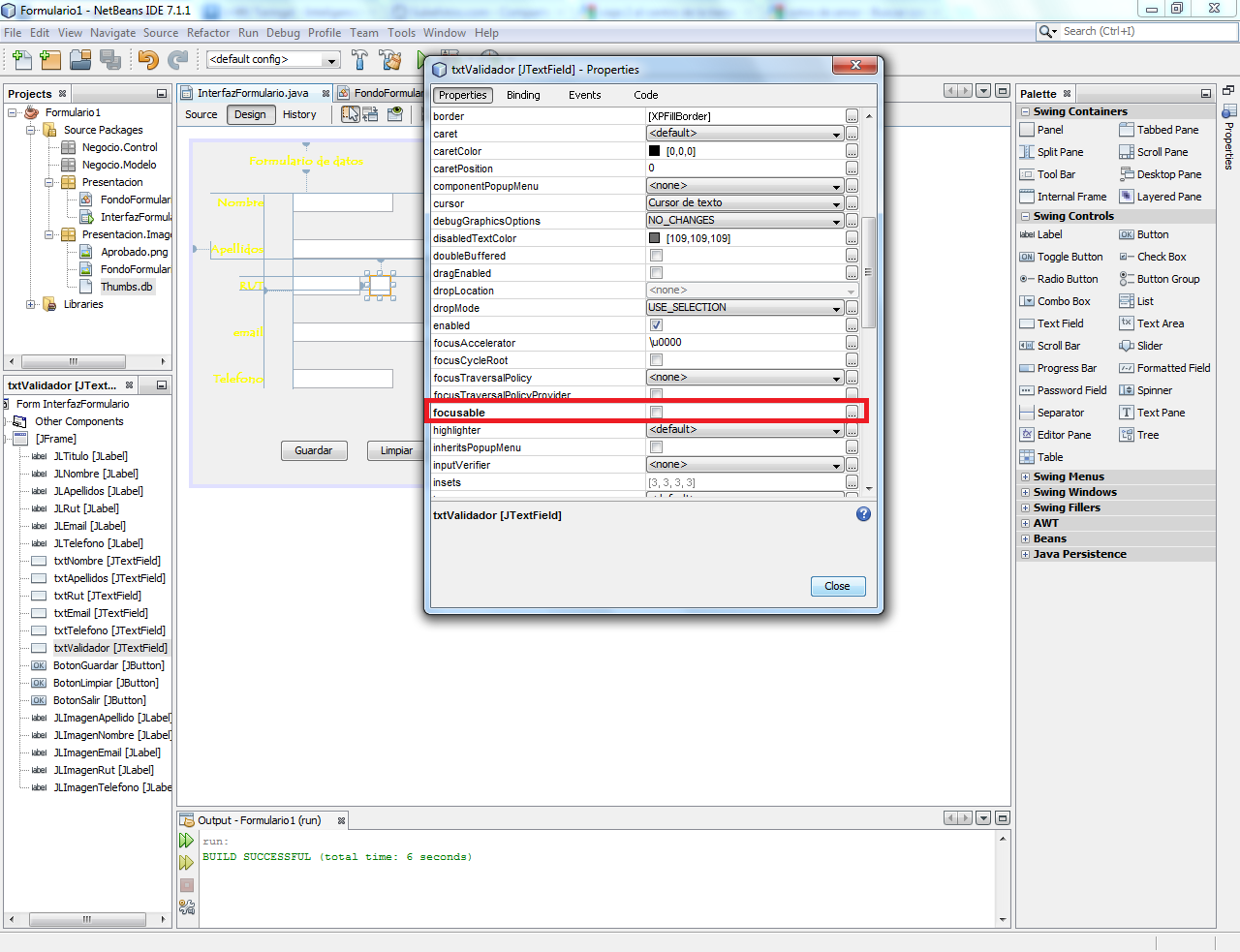
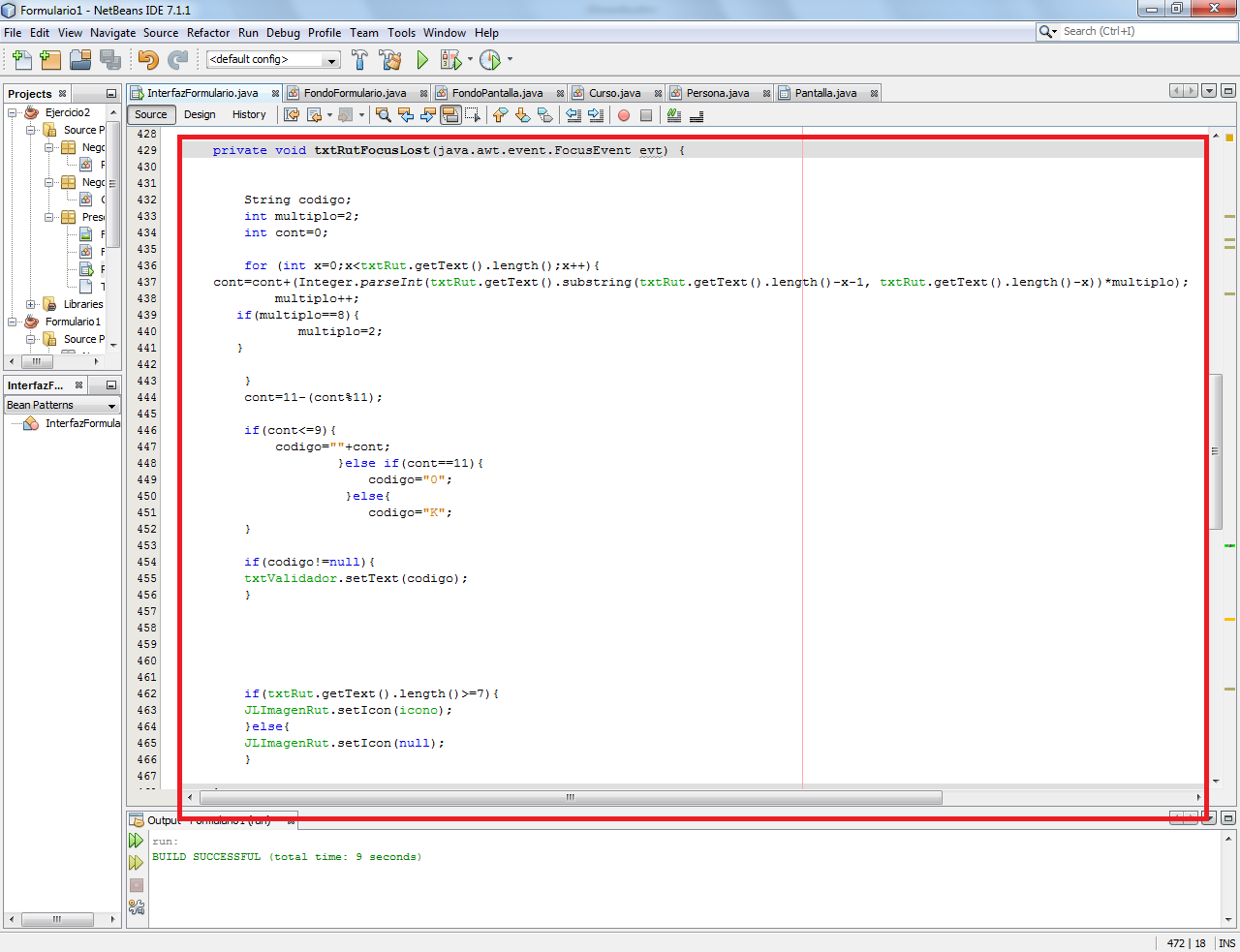
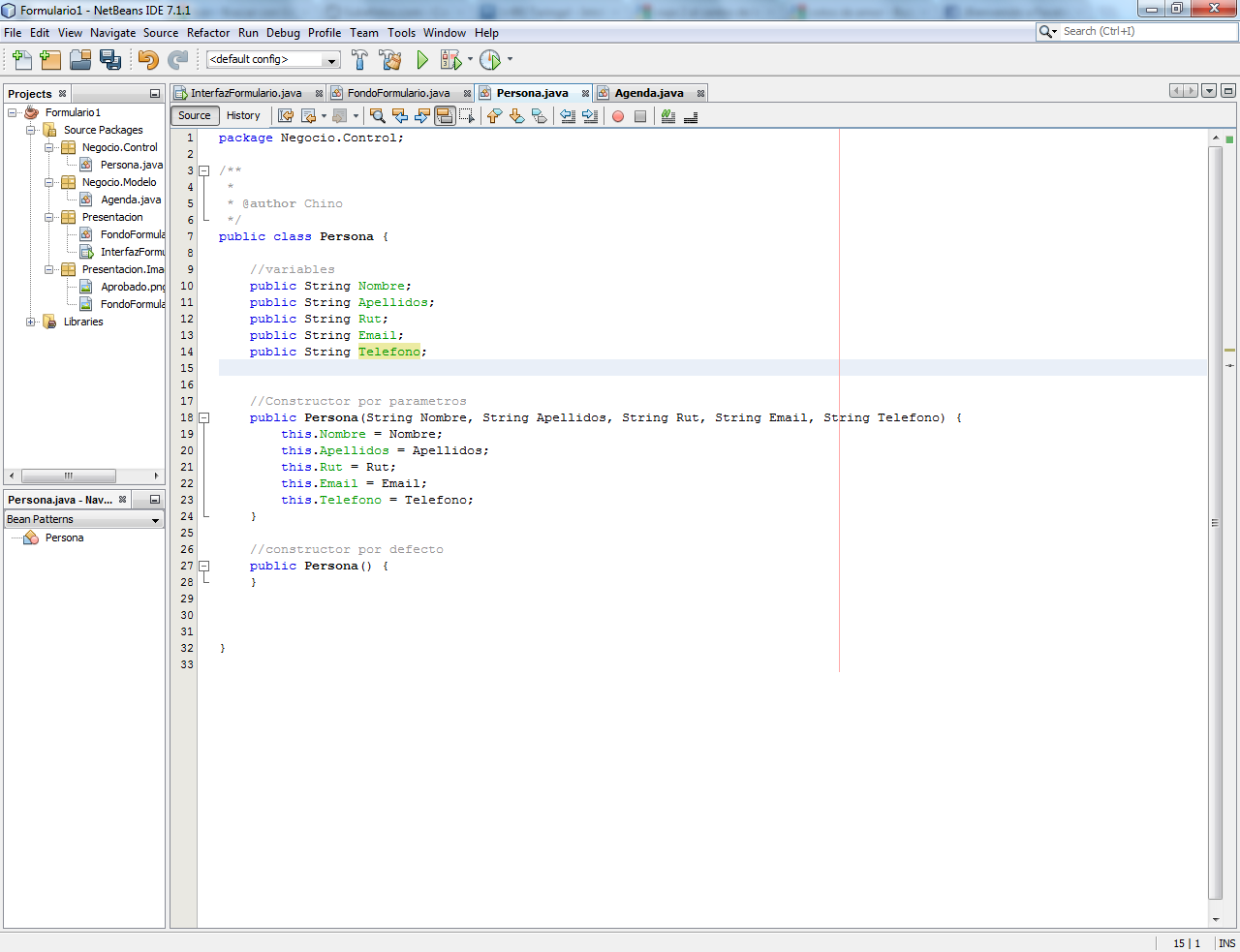
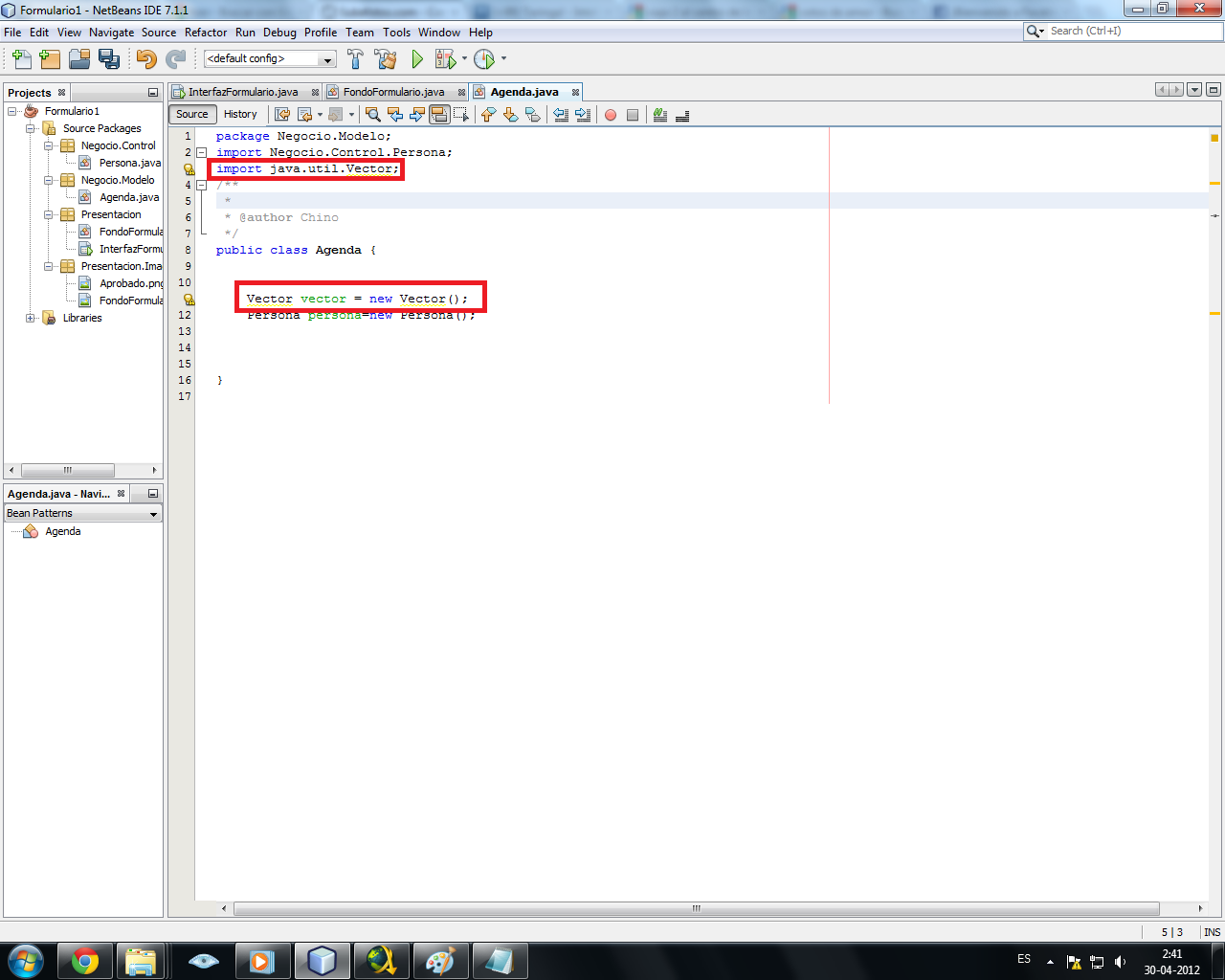
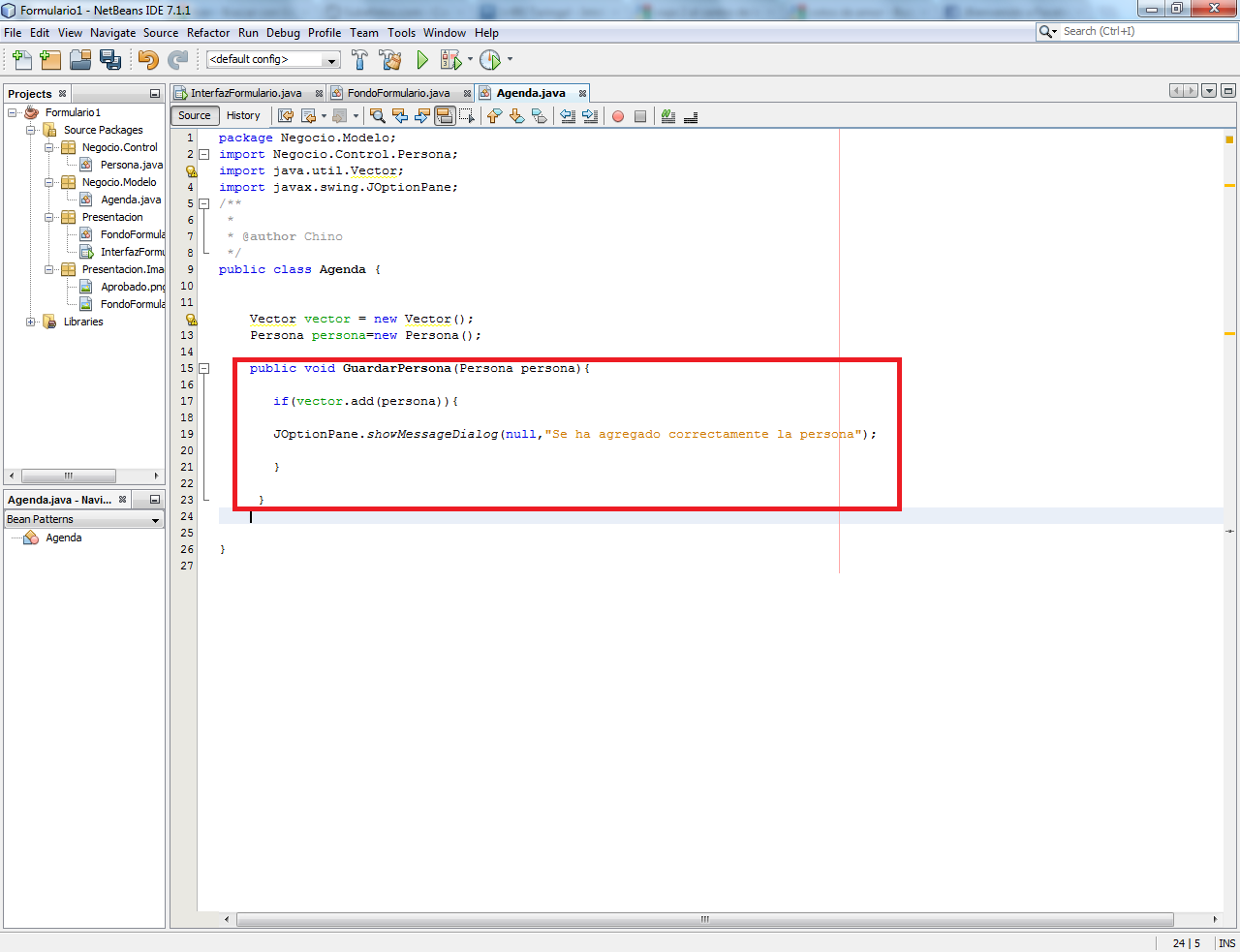
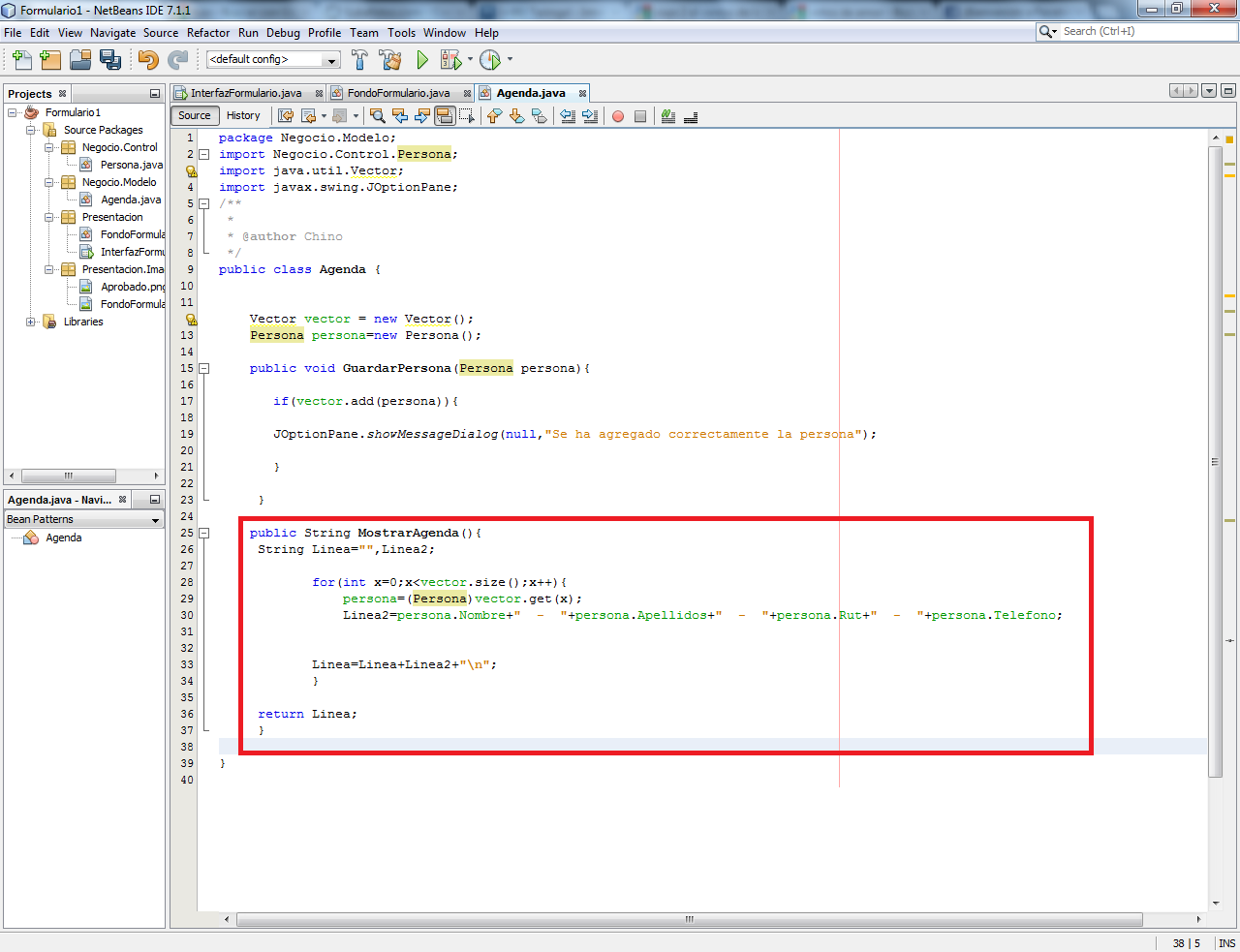
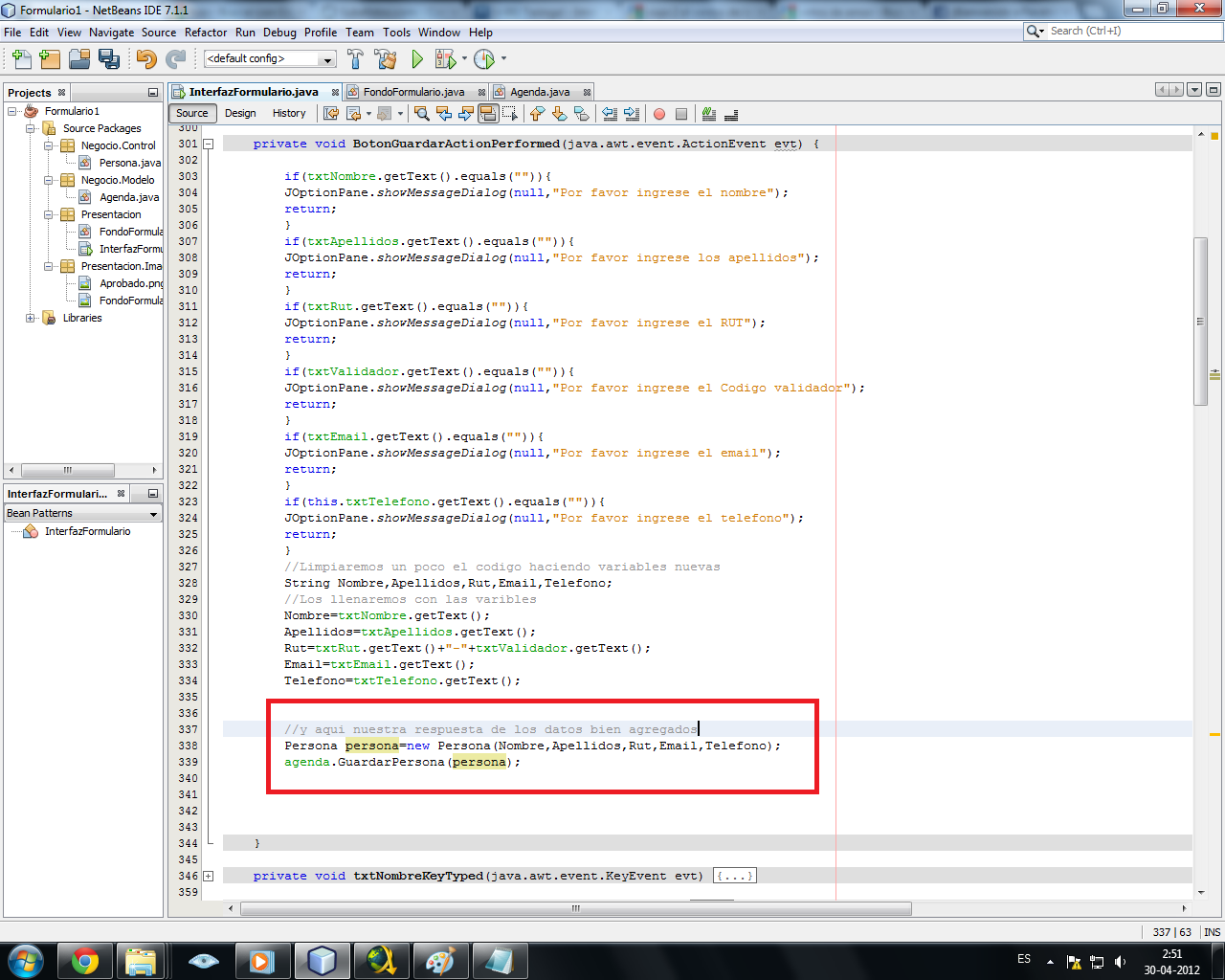
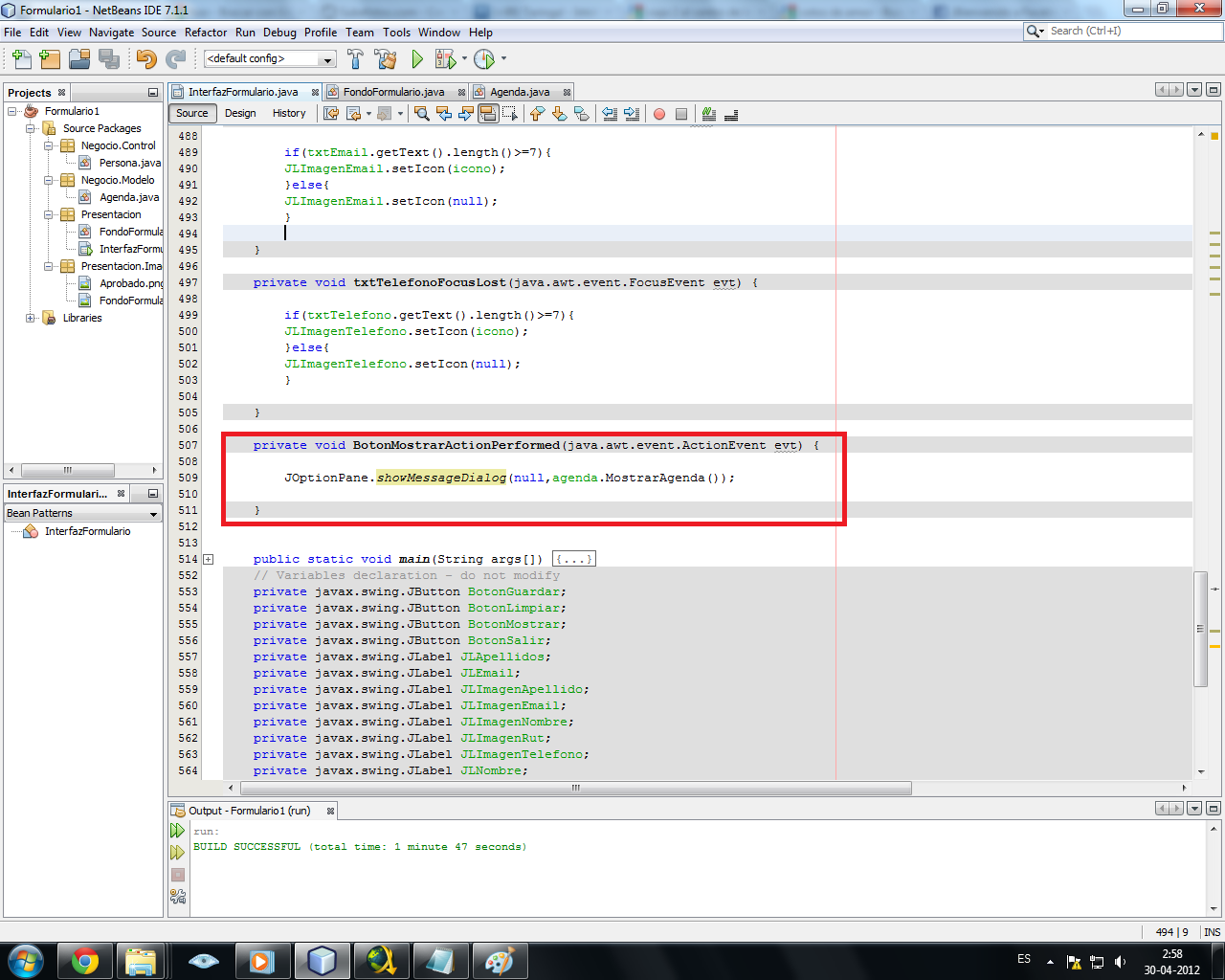
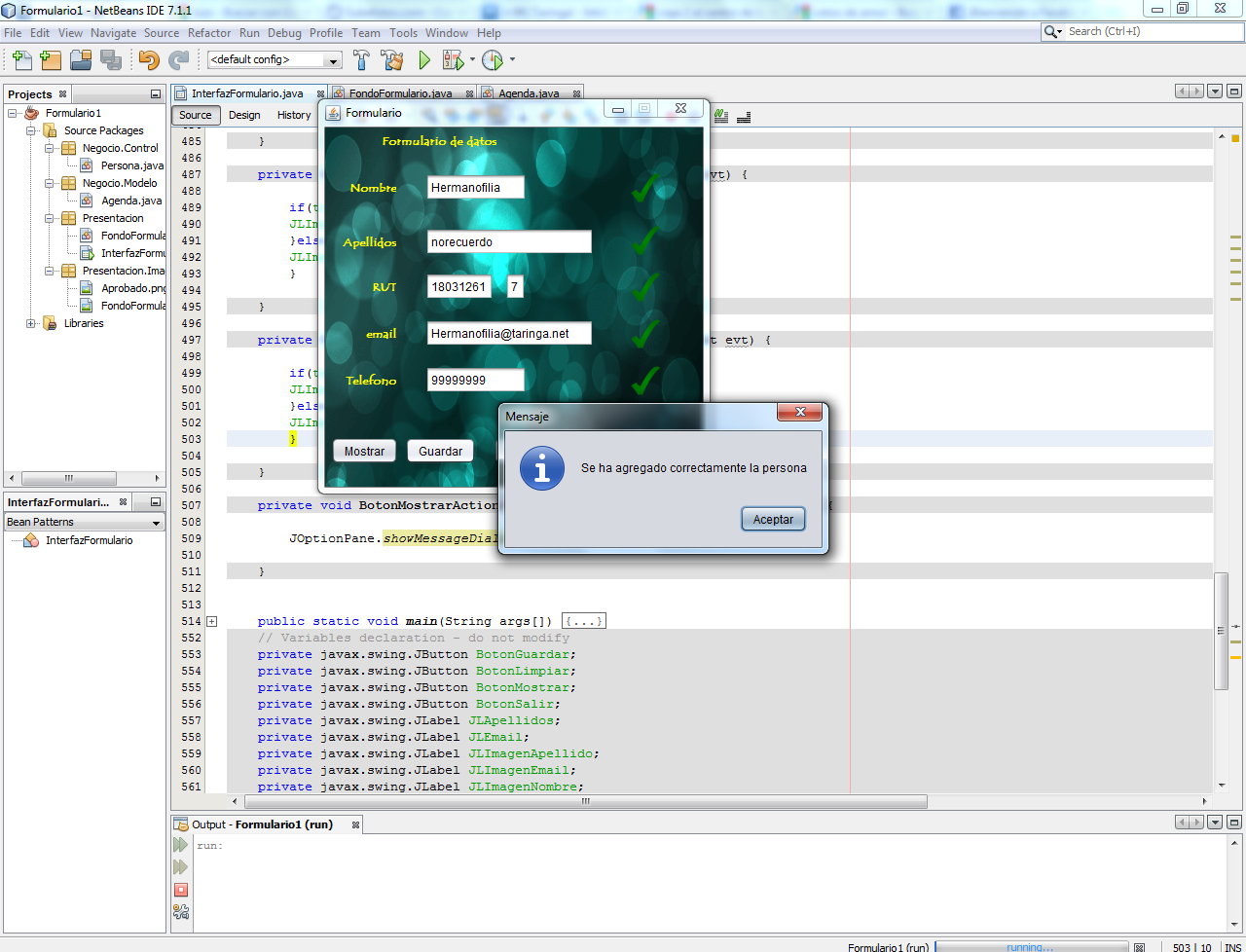
Image image = new ImageIcon(url).getImage();

@Override

public void paint(Graphics g){

g.drawImage(image, 0, 0, getWidth(), getHeight(), this);

}

}  
  
  
N°3.- Se Programara el botón Guardar con las siguientes cosas. no se guardara en ninguna parte solamente en RAM (de momento), se validara que se hallan llenado los campos obligatorios y se mostrara en un JOptionPane lo que se guardara a futuro...   
  
N°3.1.- Validar llenado de campos obligatorios   
Esto lo haremos de la siguiente manera Preguntaremos al programa si los campos están llenados en caso de que no sea así le mandamos un mensaje al usuario que diga que por favor llene el campo faltante y le ponemos la sentencia "return" para que asi no siga adelante con las demás acciones...   
  
  
  
N°3.2 Ningún nombre y ningún apellido posee números, el teléfono no posee letras (al menos en Chile no), el RUT no posee letras por lo cual aquí haremos algo interesante que solucionara algunas cosas que hay que hacer para validar números...(nos evitaremos el validarlos con un try catch que algunas veces queda mal y el error es mucho mas grande)...   
  
Diremos que el tipo de tecla que será capturado en el JTextField serán letras y cuando sea un numero diremos que eso no se escriba y que se consuma el evento... esto lo aremos de la siguiente manera...   
  
Nos posamos sobre el JTextField que queramos editar, le damos click secundario vamos a "Events" -- "Key" -- "KeyTyped"...   
ahí capturamos el evento (que es cuando el usuario apreta una tecla) en un char y lo analizamos y decimos al programa, ¿es un dígito ese char? si es un dígito entonces consumimos el evento (osea lo hacemos nulo, que no lo escriba en el JTextField), sino solo se escribe...   
(Verán como les servirá en un futuro esta parte)   
  
  
  
  
y así quedaría cuando queremos introducir solo números...   
  
  
  
N°3.3 Ahora veremos que nuestro usuario no meta demasiados caracteres y en algunos casos solo los necesarios, para practicas del ejercicio solo se hará con la longitud del texto   
por lo cual también debemos entrar en el KeyTyped contar la longitud de nuestro texto y si es mayor a la longitud que nosotros queremos dar entonces que consuma el evento...   
(se añade arriba del paso 3.2 como veremos a continuación )   
  
  
  
  
N°4 Haremos que cuando sea llenado un campo "aparezca" una imagen de un Ticket al lado haciendo semejanza de que el campo esta correctamente llenado... veamos como se hace esto...   
  
N°4.1 Crearemos JLabels del porte de nuestra imagen o del tamaño que necesitemos en este ejercicio se necesitan 5 JLabels para ingresar imagenes...   
Agrandaremos el JFrame principal para tener espacio y los colocaremos en el (los JLabels) primero los dejaremos de un porte diferente y luego borramos lo que dice... ahora le daremos click secundario 1.- cambiaremos su nombre de variable por   
ej: JLNombre --- txtNombre --- JLImagenNombre   
y así con todas los JLabels que contendrán imágenes   
teniendo esto le diremos cual sera su tamaño, el cual para este ejercicio lo dejaremos en 30 x 30 de la siguiente manera:   
  
  
  
N°4.2 Programaremos el JTextField para que cuando se rellene y el usuario pase a otra casilla salga el ticket verde...   
esto se hace con el evento llamado FocusLost   
entonces en el JTextField le damos click secundario "Events" -- "Focus" -- "FocusLost"   
En el cual preguntamos si la longitud del texto escrito es superior a 2 o 3 letras entonces que el JLabel de la imagen cambie por un ticket caso contrario que la imagen sea un null.   
lo que se hace de la siguiente manera:   
  
(primero debes iniciar una variable que contenga nuestro icono de la siguiente manera y en ese lugar dado que la ocuparemos mas de una vez...):   
  
  
  
y luego hacemos vamos a "Events" -- "Focus" -- "FocusLost"   
  
  
  
Una vez creado esto con todos nuestros JTextField tendremos que reeditar nuestro fondo dado que ahora no tiene las mismas dimensiones entonces para saber las dimensiones de nuestro JFrame principal damos doble click en cualquier borde de este y nos aparecerá un mensaje que nos dirá el largo y ancho separados por una coma...   
  
Entonces vamos al FondoFormulario.java y editamos donde dice "this.setSize(aqui damos nuestros 2 valores separados por una coma) ;   
Lo mismo abajo, PERO deben darle un poco mas de largo y de ancho para que la imagen no se vea apretada...   
(Eso lo van arreglando a medida de las ejecuciones)   
  
  
  
Ya con eso tendríamos el siguiente resultado ...   
  
  
  
BONUS TRACK CHILENOS   
  
Para auto calcular el código verificador se hace lo siguiente numero 1 se bloquea el txtValidador para que el usuario no pueda seleccionarlo   
  
  
  
Luego en FocusLost de el txtRut aremos unos ciclos que permiten calcular el código verificador   
  
  
  
Y ya esta.... Es un poco difícil de entender quizá no me especifique bien pero si lo ocuparan dirán "Que clase de brujería es esta" xD   
pero si navegan un rato por la net verán que es lo mismo que hacen todos para calcularlo   
  
Negocio.Control   
  
En esta parte crearemos una clase la cual poseerá solo atributos y en casos apartados puede tener uno que otro método entonces dentro de Negocio.Control le damos click secundario y "New" -- "Java Class"...   
Yo le llamare "Persona" y le daré los atributos que pedimos en la interfaz gráfica, are un constructor por defecto y uno con valores por parámetros, nos quedara algo así:   
  
  
  
Negocio.Modelo   
  
En este package es donde albergaremos la mayor cantidad de métodos y la menor cantidad de atributos   
Para tener todos los atributos de Persona lo aremos de la siguiente manera:   
  
Haremos una clase de nombre Agenda ( Negocio.Modelo le damos click secundario y "New" -- "Java Class"...)   
  
Ahora lo que resta es hacer Metodos para utilizar los atributos y luego implementarlos en la Interfaz   
Así que crearemos el método guardar, el cual nos guardará en un arraylist o un vector o como le llamen...   
se hace así   
  
(creación de un vector importando la clase java.util.vector)   
  
  
  
Por consiguiente debemos llenar ese vector de Personas, pero los vectores se auto crean con capacidad de llenarse con Object (es por defecto del que hizo ese código) por lo que una ves que yo entregue al vector una clase de tipo Persona el la guardara como tipo Object por lo cual al sacar nosotros la información debemos devolver el Object a Persona... (si les resulta estresante o enredado es cosa de ver el código y lo entenderán)   
  
N°1.1.- Guardar:   
En este método lo que se hará es Guardar lo que tiene Persona en el Vector de la siguiente manera   
  
  
  
N°1.2.- Mostrar Agenda   
En este método lo que se hará simplemente es en una variable llamada "Linea" albergaremos las lineas, valga la redundancia, de todas las personas en el vector (recordar que como todo lo que hay en ese vector es de tipo Object hay que devolverlo a Persona antes de poder ocuparlo)   
  
  
  
Muy Bien con eso tenemos lista la parte de Negocio http://o1.t26.net/images/big2v5.gifhttp://o1.t26.net/images/space.gif   
  
Ahora solo resta juntar Negocio con Presentacion y el código estará completo   
  
Lo que haremos es mostrar la agenda en un JOptionPane para no tener que desarrollar un JFrame nuevo o meterle un JtextArea al Frame ya hecho.   
  
nos dirigimos al JFrame principal y al boton Guardar le llevaremos el metodo Guardar que acabamos de crear de la siguiente manera   
  
Modificar Guardar...   
  
  
  
Crear un BotonMostrar para ver si se guarda el curso completo en nuestro Vector   
  
  
  
Y los resultados que tiene nuestro Programa son....   
  
Resultados   
  
(Guardar Exitoso)   
  
  
(Mostrar Exitoso)   
