**MVC**

MVC (**Modelo-Vista-Controlador**) es un patrón en el diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. Enfatiza una separación entre la lógica de negocios y su visualización.

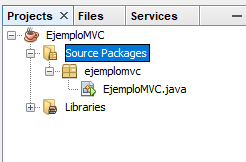
Este es el patrón que utilizaremos de ahora en más para trabajar con nuestros proyectos en Swing y Base de datos.

Primero que todo, instalaremos netBeans 8.2, para esto iremos al siguiente link:

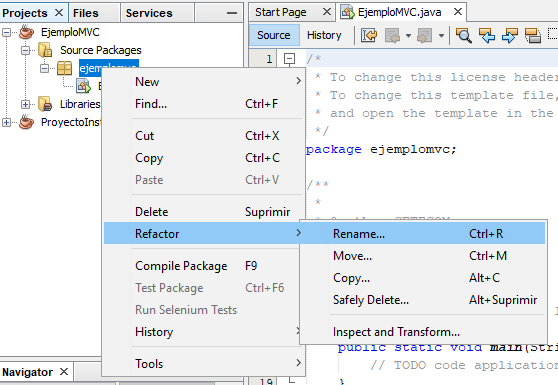
<https://drive.google.com/file/d/1-MpuBVP22gA5UKzaiYn-j1R5OcOSymcq/view?usp=sharing>

Descargaremos el instalador y lo instalaremos en nuestro equipo, usaremos este porque no nos dará problemas para trabajar con Base de datos.

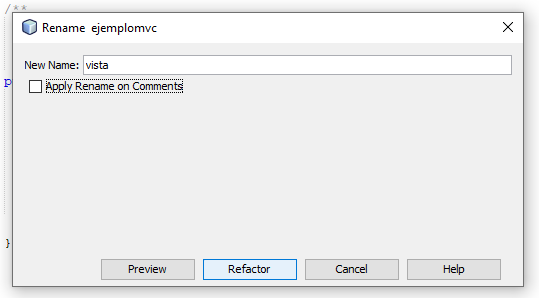
Crearemos nuestros proyectos y separaremos nuestros packages de esta manera. Para el ejemplo, crearemos un proyecto al que llamaremos EjemploMVC



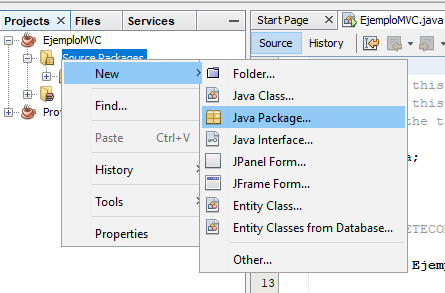
Quedaría de esta manera, nuestro main (aplicación), con la que hemos trabajado hasta ahora, deberá quedar dentro del package vista, para esto cambiaremos el nombre al package “ejemplomvc”, botón derecho:



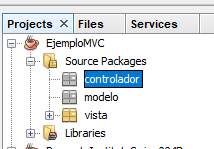
Y le ponemos como nombre a ese package **vista**, luego, de poner el nombre, hacer clic en refactor



Después crearemos un package llamado **modelo** y otro llamado **controlador**

****

Y nos quedaría



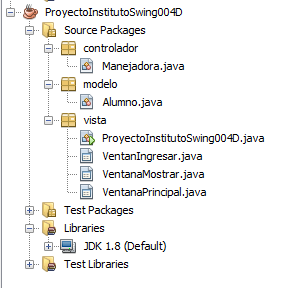
Los packages modelo y controlador aparecen en gris porque aún no tienen algo dentro de ellos.

En el package modelo irán las clases que utilicemos.

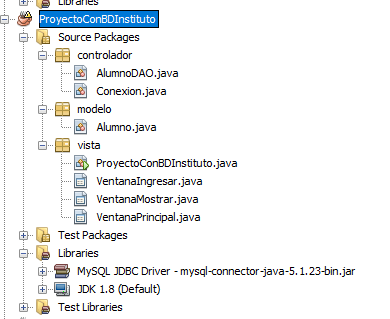
En el package controlador ira la manejadora, y si trabajamos con base de datos, la clase de conexión y la manejadoraDAO.

En vista, además de nuestra aplicación, irán las ventanas de swing donde programaremos.

Ejemplo de un esquema sin base de datos:



Ejemplo de un esquema con base de datos:



En este se incluye la clase Conexión y la clase AlumnoDAO, Y también en Libraries MySQL JDBC Driver.

**Swing java**

Swing es una biblioteca gráfica para Java. Incluye widgets (pequeña aplicación o programa) para interfaz gráfica de usuario tales como cajas de texto, botones, listas desplegables y tablas, entre otros.

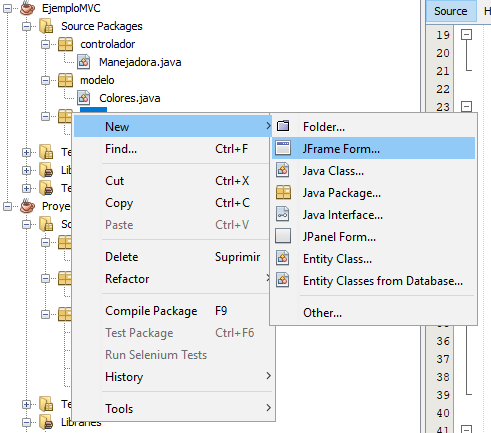
Volvamos al ejemplo que teníamos hecho de MVC

Crearemos una clase a la que llamaremos Colores, que tendrá tres atributos, un código, una cantidad, y un color. Recuerde que la clase va dentro del package modelo. Recuerde su constructor, get y set.

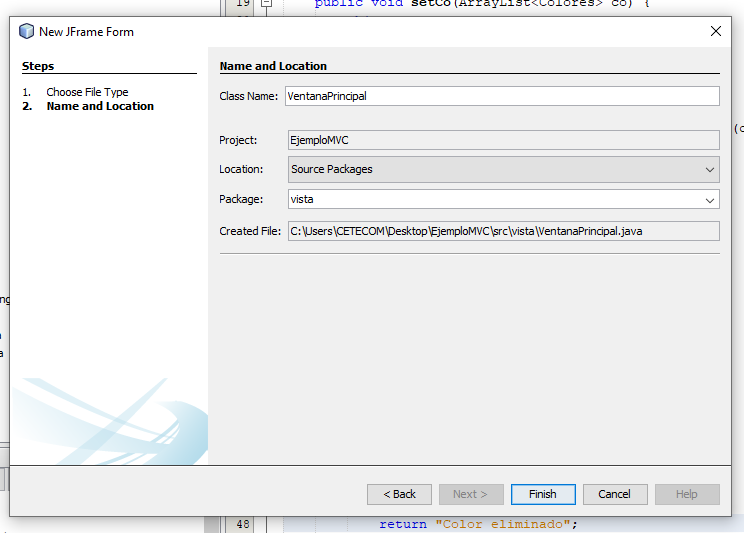
Una manejadora que irá dentro de controlador, que tendrá un ArrayList de tipo Colores, que permitirá ingresar, buscar y eliminar. En la manejadora, además de pedirme importar el java.util.ArrayList, también me pedirá importar lo que tiene el package modelo, porque mi clase Colores al estar en un package distinto, el no lo reconoce y necesito darle el camino de donde debe ir a buscarla.

Una vez que tengamos lista la manejadora iremos a nuestro package vista, donde tendremos el main. Allí crearemos nuestras ventanas.

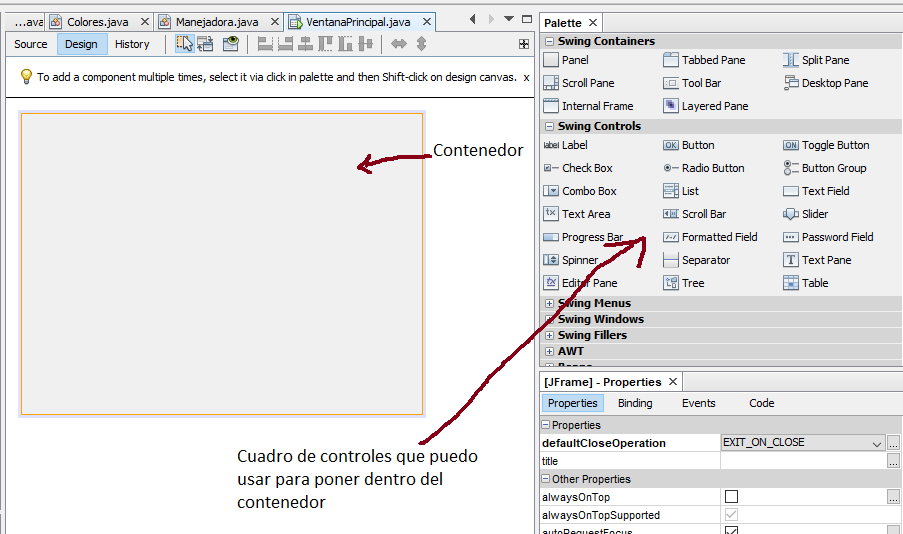
Sobre el package vista, botón derecho.



JFrame Form, eso es lo que crearemos, le daremos un nombre que comenzará con mayúsculas igual que una clase, en este caso le pondremos “VentanaPrincipal”.



Y finish, nos quedará lo siguiente:



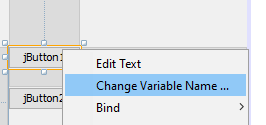
Existen diversos tipos de controles que puedo utilizar para trabajar sobre mi ventana, dentro de los más utilizados están los siguientes:

jButton  es un botón en el cuel se suele hacer clic esperando que haga algo al presionarlo.

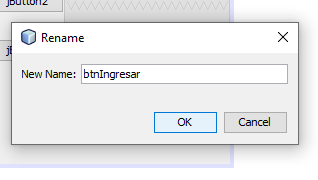
jTextField  es un cuadro de texto que se utiliza para que el usuario ingrese datos e interactúe con nuestro programa

label o etiqueta  es una etiqueta que generalmente acompaña al cuadro de texto explicando que debe ingresar en él.

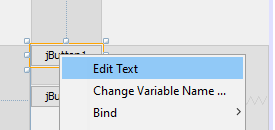
En nuestra ventana principal agregaremos tres botones, un botón que va a ingresar, otro que va a mostrar y otro que va a salir, java asigna números a cada control que agrego para diferenciarlos uno de otros, nosotros le pondremos nombres. Como se trata de botones le pondremos btnIngresar, btnMostrar y btnSalir. Para esto me pongo sobre el control que ya arrastre encima del contenedor, presiono el botón derecho y me sale un menú:



Allí le pongo el nombre btnIngresar al primero



Hago lo mismo con todos los botones. Si quiero cambiar lo que dice cada botón. Hago lo mismo pero debo elegir:



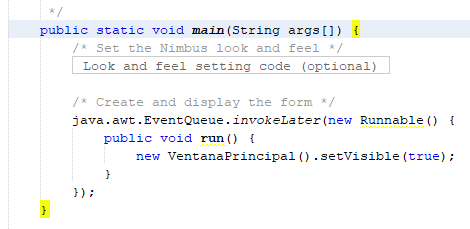
Y me quedaría esto:



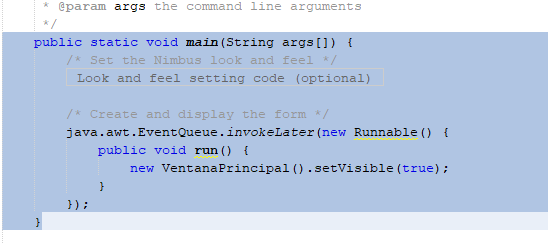
Si yo hago clic en la pestaña source de arriba 

Podré ver la codificación de la ventana donde estoy trabajando.

Si se fijan, la VentanaPrincipal y el main, tienen ambos, el símbolo verde del play, eso significa que ambos tienen un main,  lo que haremos es en la ventana de source, buscaremos el main, debemos tener cuidado, marcaremos donde comienza y donde termina para no equivocarnos.



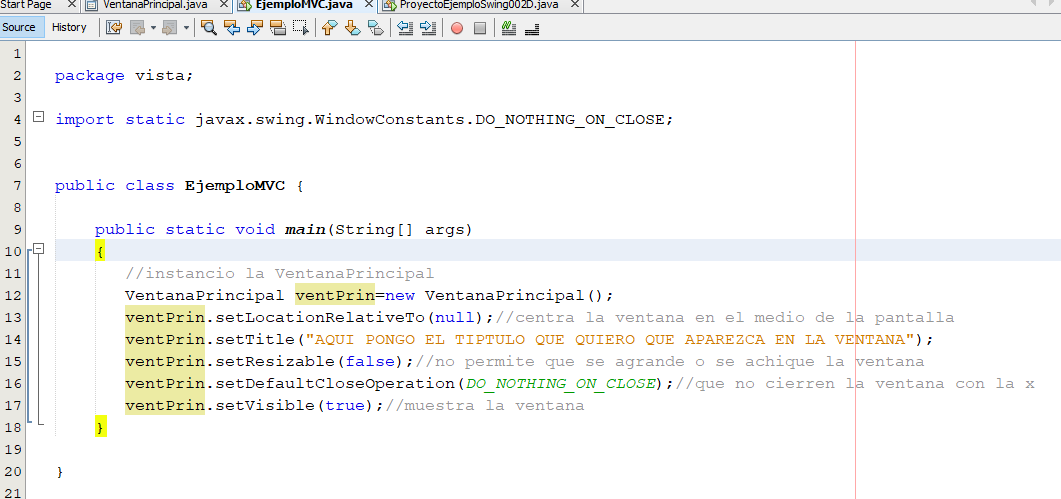
La llave me indica donde comienza y donde termina, luego seleccionamos todo el main.



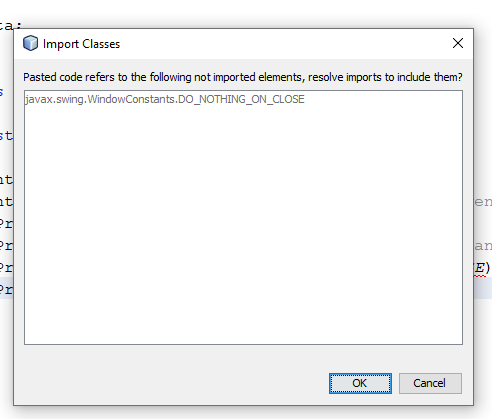
Y lo eliminamos, guardamos los cambios y de inmediato vemos como cambio la VentanaPrincipal  dejo de tener el main y ahora solo partiremos de nuestra aplicación. Eso lo haremos para cada ventana nueva que agreguemos.

Si nosotros ejecutamos nuestro programa no hace nada porque aún no hemos programado la presentación de la ventana que hemos insertado, para presentar la ventana, debemos ir a nuestra aplicación.

Nuestra ventana es una clase y como tal, necesitamos instanciarla para poder utilizarla y esto lo hacemos de la siguiente manera:



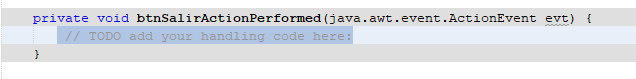
Cuando escribes estas instrucciones, java te pedirá que importes lo siguiente



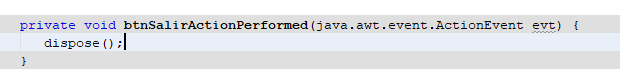
Antes de probar si funciona, necesitamos programar el botón salir.

Iremos a VentanaPrincipal, haremos clic sobre la pestaña desing 

Luego seleccionaremos el botón salir y haremos doble clic sobre el, nos encontramos dentro de este evento

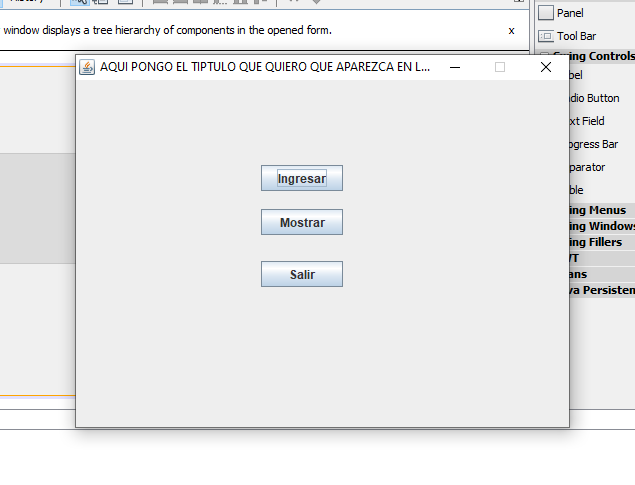


Borramos lo que tiene como comentario y escribimos dentro



Esto permitirá que podamos salir de nuestro programa.

Ahora le damos play a nuestro programa y podemos ver nuestra ventana con nuestros botones.



Ahora ven que, si trato de cerrar la ventana con la x, no me deja, tampoco me deja moverla, o cambiarle el tamaño, solo puedo cerrar el programa con el botón salir.

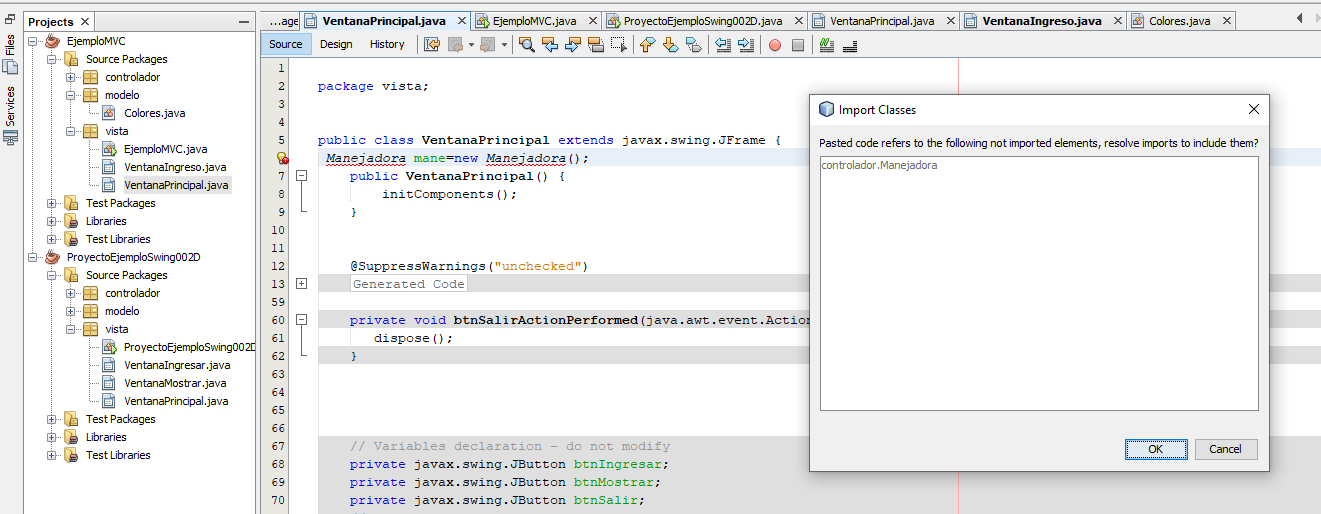
Bien, la idea es que cuando presione el botón ingresar me lleva a otra ventana donde pueda ingresar los datos de mi clase Colores. Debo crear otra ventana y en esa ventana, pedir los datos que necesito.

Agregaremos una nueva ventana a la que llamaremos VentanaIngreso, recuerden hacer lo mismo que hicimos con la principal, quitarle la codificación del main.

Pondremos en ella 3 labels uno que dirá Código, otro Color, otro Cantidad, 3 TextField, esos serán para el ingreso de datos y a esos si les cambiaremos el nombre, como serán textos, les pondremos txtCodigo, txtColor y txtCantidad, los agrandaremos dejándolos del mismo tamaño a los 3, editaremos con el botón derecho cada uno, borrando lo que trae dentro para queden en blanco. Pondremos dos botones uno dirá ingresar y el otro dirá volver, el que dice volver deberá llamarse btnVolver y el otro, btnIngresar. El botón volver lo programaremos igual que el anterior con dispose();

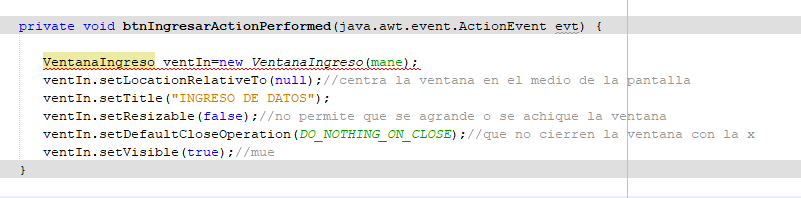
Ahora debemos volver a la VentanaPrincipal, de esa ventana llamaremos a la VentanaIngresar, cuando se haga clic en el botón ingresar queremos que aparezca la VentanaIngresar.

Bien, como estamos trabajando con una manejadora, necesitamos en esta ventana instanciar nuestra manejadora y mandar esta manejadora como parámetro a la VentanaPrincipal, de la siguiente manera:



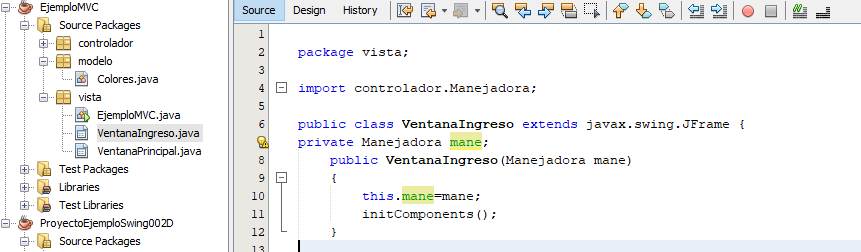
Al hacerlo me pide que importe la manejadore de controlador, le digo ok.

Luego hago doble clic en el botón ingresar de la VEntanaPrincipal, borro lo que aparece allí y codifico lo siguiente:



Debo hacer lo mismo que hice para la VentanaPrincipal, pero allí me muestra un error y es porque le estoy mandando la manejadora y en mi VentanaIngresar no estoy recibiendo nada, ya lo vamos a solucionar.

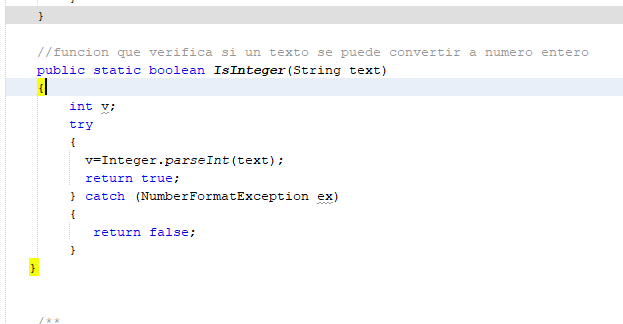
Vamos a nuestra VentanaIngresar. En esta ventana vamos a definir nuestra manejadora como si fuese un atributo de esta clase, tenemos que importar la manejadora del controlador.



Y ya no hay error.

Ahora ingresaremos datos. En el botón ingresar haremos doble clic, y comenzaremos a codificar

Vamos a crear una función para ver si un texto se puede convertir en entero y lo pondremos en esta ventana, pero fuera de la llave del ingresar.



En el botón ingresar haremos lo siguiente:



Aquí igual que en las clases, el get es el que muestra y el set es el que modifica. requestFocus() manda el cursor al control designado.

JOptionPane tiene muchas formas de utilizarse, investigue cuales mas puede utilizar.

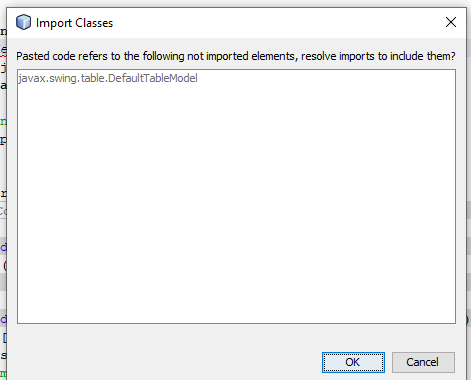
Bien ya hemos ingresado datos, ahora mostraremos datos en otra ventana, Debemos crear una ventana llamada VentanaMostrar, tenemos que hacer lo mismo que hicimos para las otras dos.

Esta ventana recibirá la manejadora también igual que la ingresar.

Mostraremos en una tabla para que se vea mejor y antes de definir la manejadora pondremos lo siguiente:

DefaultTableModel modelo= new DefaultTableModel();

Usaremos ese modelo para trabajar, luego y me pedirá que importe

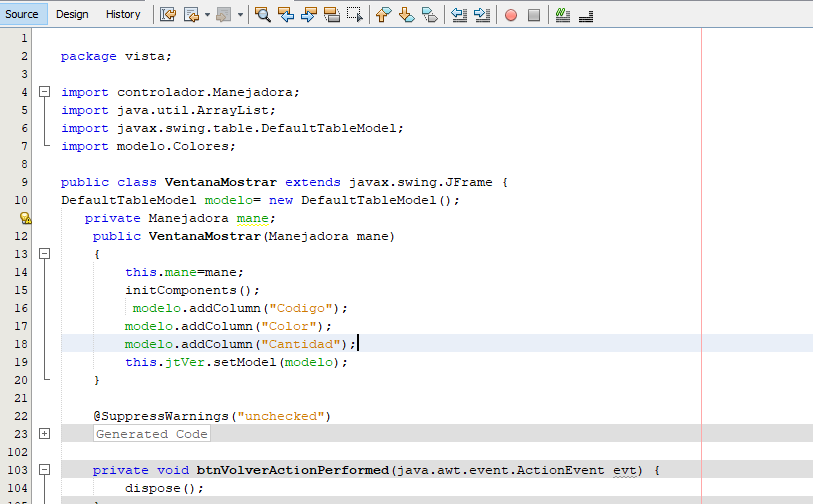


Agregaremos a la ventana mostrar un jTable y le pondremos de nombre jtVer 

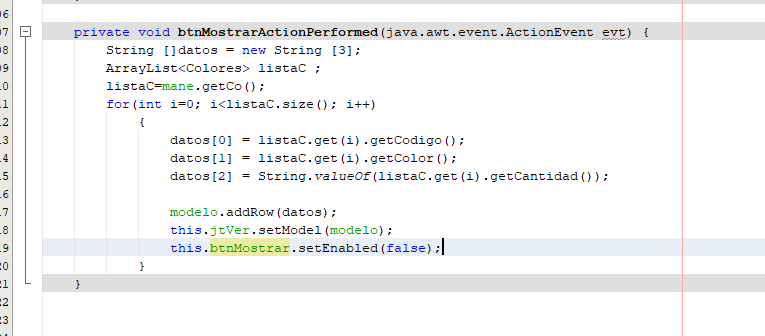
Luego nos ponemos sobre el control, botón derecho, table contetens y allí puedo decirle cuantas columnas quiero, que dirán las columnas, cuantas filas, etc.

Agregamos 2 botones, uno que diga mostrar todo y otro volver.

La ventana mostrar quedará al inicio de la siguiente manera:



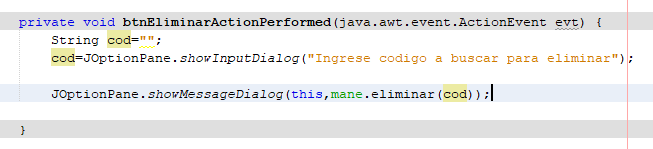
En el botón mostrar todo codificaremos lo siguiente:



Tenemos que presentar la VentanaMostrar igual como lo hicimos con la anterior desde la ventana principal.

Por último agregaremos un botón más a la ventanaPrincipal y a ese botón le pondremos eliminar en el texto y de nombre btnEliminar.

Dentro de la codificación:



Y ya tenemos un pequeño proyecto listo y funcionando.

