Mysqul workbench con Net Beans para crear un base de datos

* Poner
* Insertar
* Buscar
* Mostrar
* Eliminar
* Modificar
* Eliminar
* Modificar

1.- Abrir MYSQL Worbench

Se le asigna un nombre a la conexión de la base de datos (CONEXION) ya que es importante asignarle el nombre. Se abre la nueva página para iniciar el código, se le asigna el nombre que se le asignó a la base de datos (CONEXION).

Se empieza a redactar el código que se va a utilizar. Se crea la base de datos llamada CONEXIÓN y se crea una tabla.

Es importante que el nombre de la conexión y de la tabla no sea iguales, ya que no se pueden repetir las variables que se han utilizado.

El INT plus se utiliza para agregar números y el varchar para agregar letras y números, en el varchar (50) es el número máximo de caracteres se pueden utilizar.

Se utiliza el rayo (ϟ) para compilar el código y verificar en la actualización de los schemas.

Al compilar (ϟ) no marca ningún error porque se acaba de crear.

Actualizar el código aparece en los schemas, para en el momento de conectar la base de datos con Net Beans no se encuentre ningún problema.

Conectar la base de datos

Para la conexión de la base de datos no se le asigna ningún nombre se deja vacío.

Después se abre una nueva página, donde indica que el procedimiento anterior está correcto.

Se abre Net Beans

Se crea un nuevo proyecto con el mismo nombre que fue asignado a la base de datos (conexión). Aparecen las librerías y se agrega nueva enlazada a MYSQL JCBC DRIVER.

En la parte de services se selecciona la Data base y se conecta MYSQL.

Se busca la conexión creada.

Al conectar pide una contraseña (password) se deja vacío o cuando tienes una cuenta se agrega la contraseña que usas.

Se crea nuevamente 2 cuestiones, 1 se de JAVA CLASS se le asigna el nombre de CONECTAR, la otra es de JFRAME FROM no importa el nombre que se le asigna.

En el new JFRAME.JAVA se crea

1.- se inserta 2 LABEL totalmente opcionales y un título de conexión.

Es importante agregar 2 TEXTFIDEL y un botón.

Conectar JAVA CLASS

Dentro del código que se creó se tiene que asignar el nombre de CONEXIÓN por la base de datos.

Se realizan dos asignaciones

* Insertar
* modificar

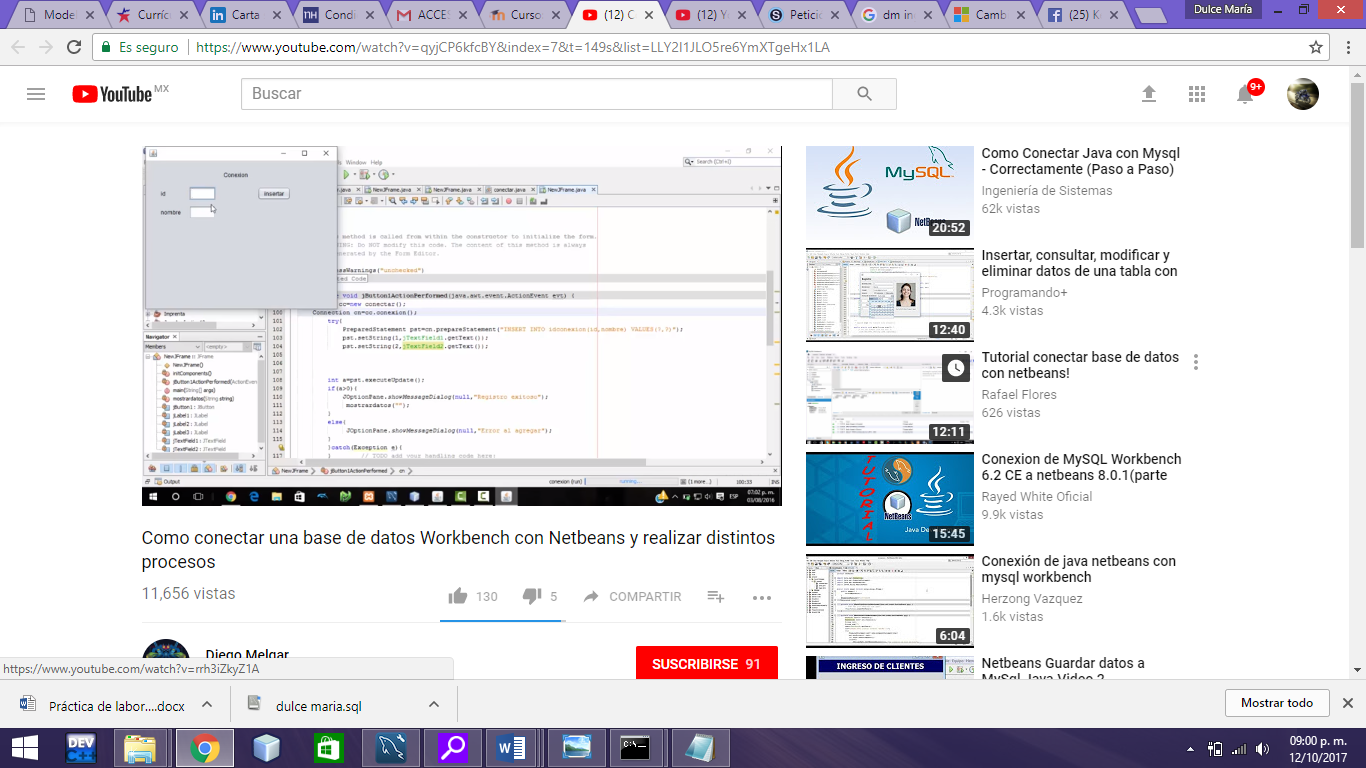
Se introduce el código en el boto de insertar

1.- checar los TEXFIDEL que estén conectados y comprobar con las LABEL

2.-checar los id\_conexion, id\_conexion.

3.-checar el error y solucionarlo en créate method.

Se corre el programa en RUN FILE. Y aparece una nueva ventana



Implementar los botones

* buscar
* modificar

Para ellos se crea 4 botones

* 2 botones
* 1 JTextFidel
* 1 JTable

El botón de mostrar sea llamado JButton2 y el de buscar JButton3 y otro se llama JTextFidel3.

Se utiliza el código que involucrar mostrar y buscar, el otro no será programado en ninguno de los dos botones será implementación en el código (PUBLIC NEW J FRANCE).

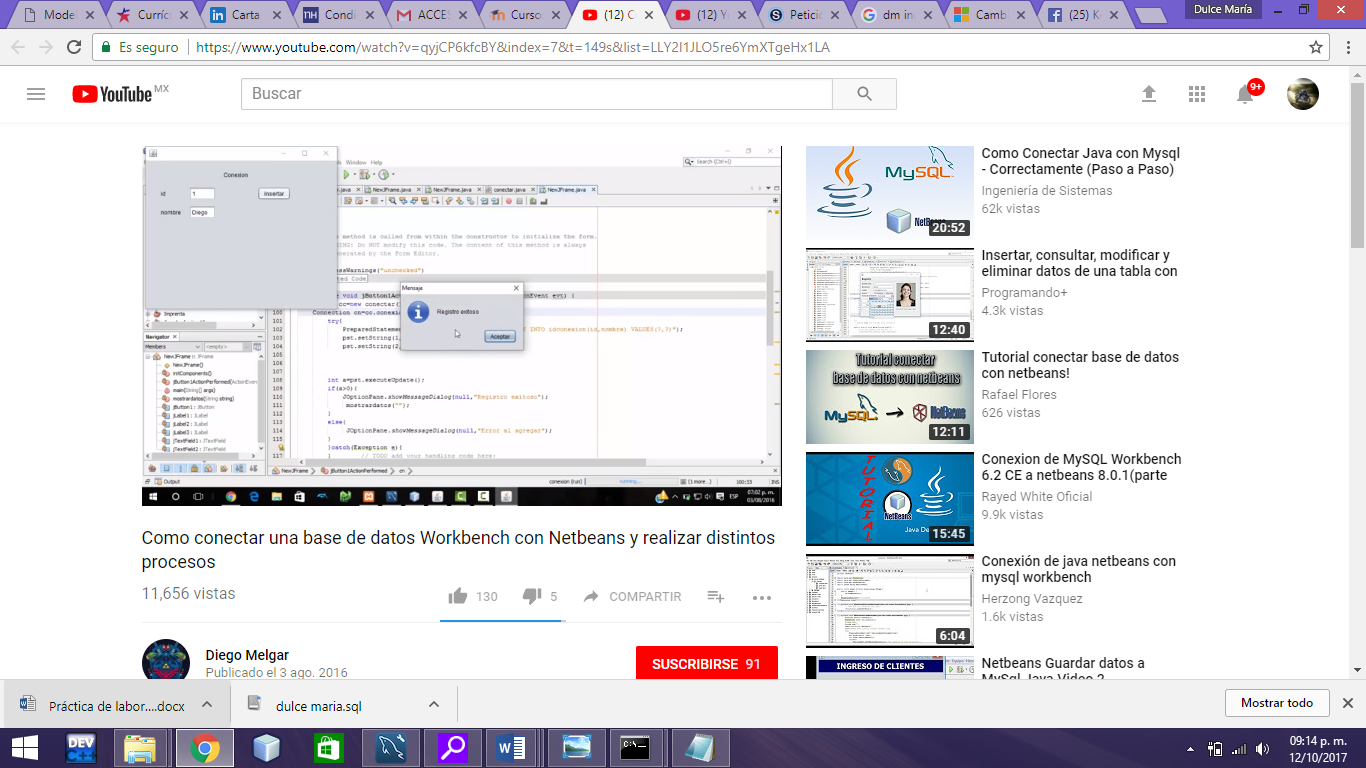
E corrige un error y al solucionarlo crea un segundo diálogo en PRIVATE VOID.

Se tendrá que eliminar uno de los diálogos para que no esté repetido y esté libre de errores.

Implementar

En SELECT FROM se selecciona la tabla = id\_conexion

En id y nombre se busca como un número y un nombre.

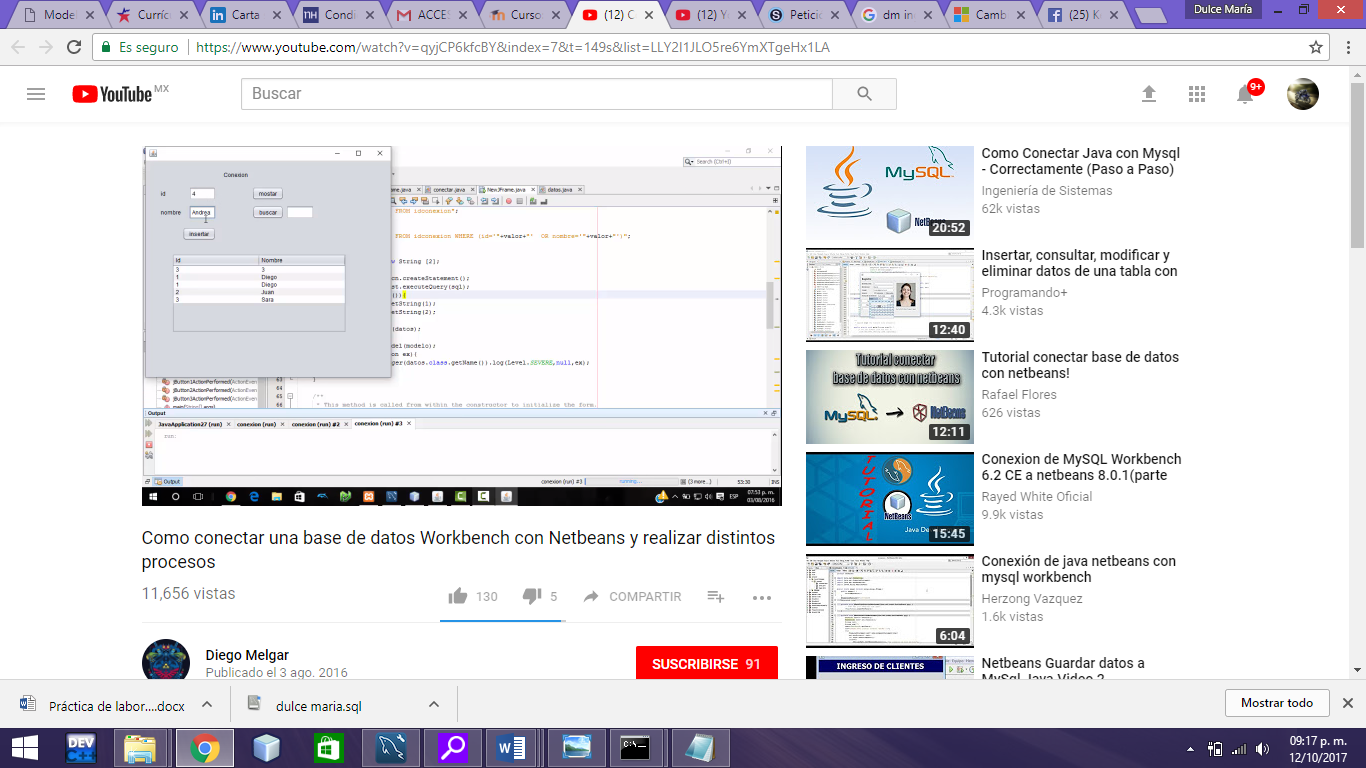


Línea de código en el botón mostrar=mostrar datos (““); y el botón buscar.

Cambiar nombre de la tabla, id\_conexion.

Verificar todo el programa.

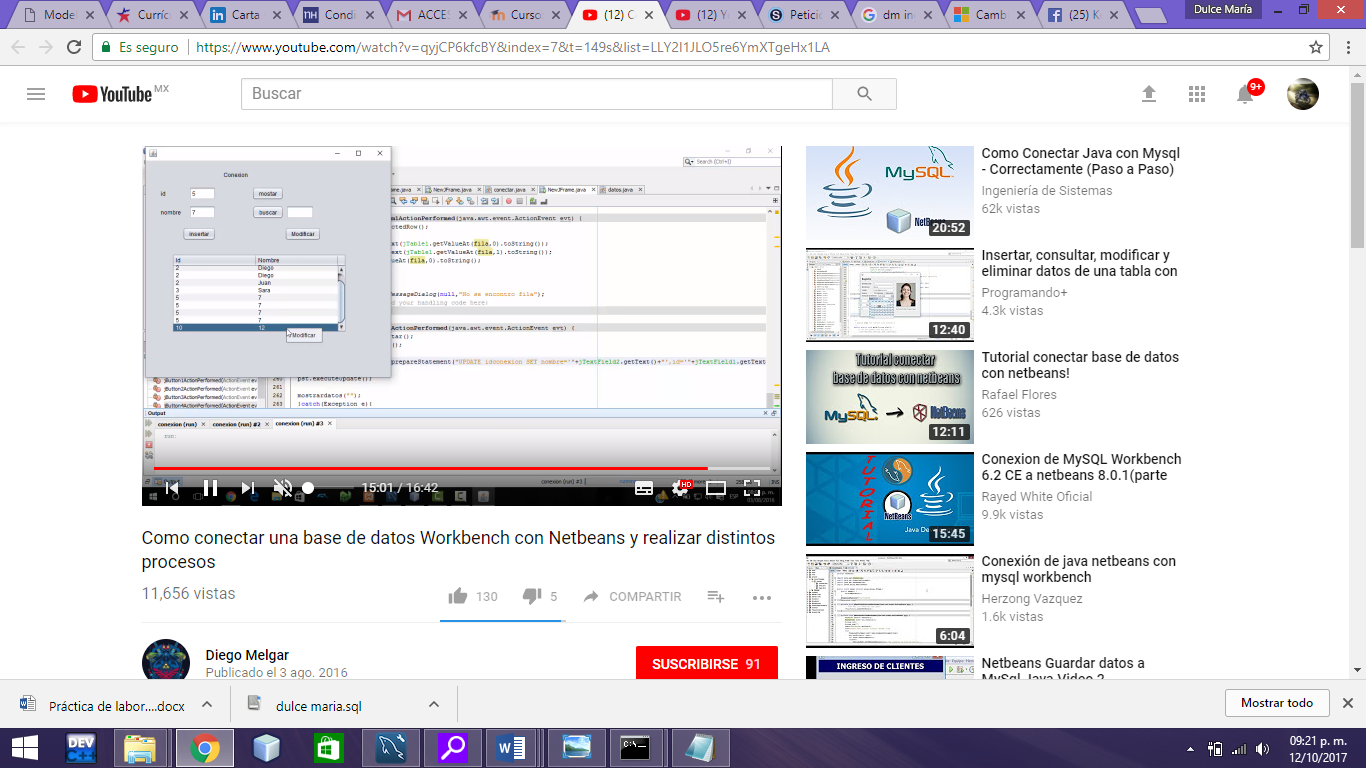
Mostrar tabla



* Eliminar
* Modificar

Modificar

Agregar un botón llamado POPUPO MENU, se modifica el texto.



Código implementar en JMENUITEM1 (JMenultem)

Se selecciona y aparece un código.

Corrección de errores.

Verificar los JTexFieldl

En el botón de modificar va un código importante pero marca error en id.

Se crea las variables STRING para solucionar el error.

STRING id=” ”; de esta manera se corrige todos los string.

Si se modifica el nombre no afecta string ayuda a evitar las modificaciones globales.

Eliminar= JPopupMenul(JPopupMeNU), se selecciona y se modifica sus propiedades.

Test-Eliminar

Se inserta el código de descripciones

Se cambia el nombre de la tabla a CONEXIÓN.