

IMPLEMENTACIÓN DEL MODULO DE INICIO DE SESIÓN Y GESTION DE PANEL INICIAL CON JDBC

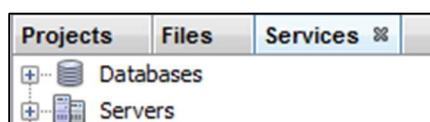
LOGRO DE SESIÓN: Al finalizar la sesión el estudiante es capaz de implementar el módulo de inicio de sesión y gestión de pantalla principal de sistema de información.

ACCESO A LA BASE DE DATOS DESDE NETBEANS

NetBeans nos permite hacer operaciones sobre la base de datos como crear y borrar tablas, agregar y eliminar columnas, agregar, modificar y eliminar registros de datos como realizar consultas.

Vamos a proceder hacer algunas operaciones:

1) CONFIGURACIÓN PARA ACCEDO A DATOS DESDE NETBEANDS.



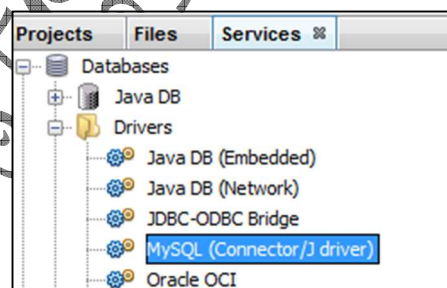
A) En el Explorador de proyecto Accede a la ficha “Servicios”.

B) Expandir el icono **Databases**, Drivers.

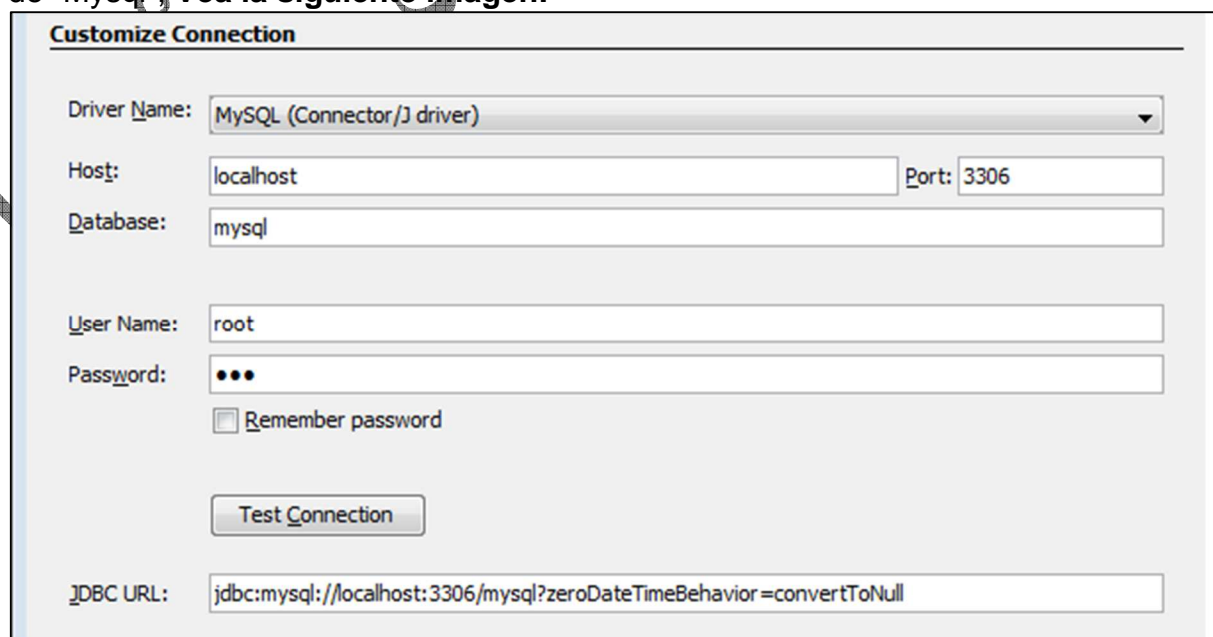
C) Clic Secundario en el Driver:

“MySQL(Connector/JDriver)”

D) Elegir la opción “Usar Conexion”



E) En el asistente de configuración ajustar algunas opciones como contraseña de “Mysql”. **Vea la siguiente imagen:**



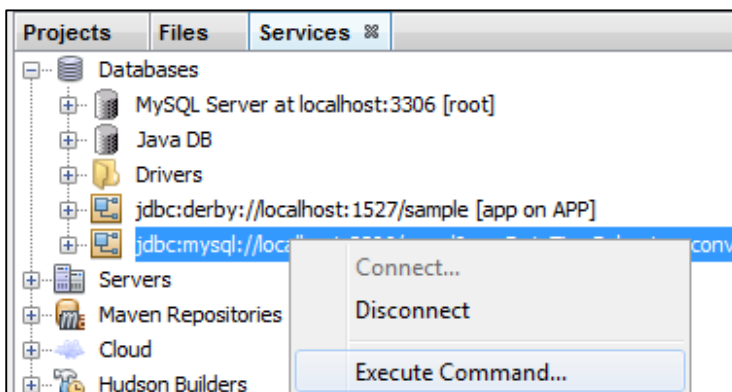
F) Finaliza la configuración.



2) ELABORAR UNA BASE DE DATOS EN MYSQL DESDE NETBEANDS

- A) Clic Secundario en la Conexión Creada, Seleccionar la opción "Execute Command".

IMPORTANTE. Debe abrir una ventana que servirá para escribir "CODIGO SQL"

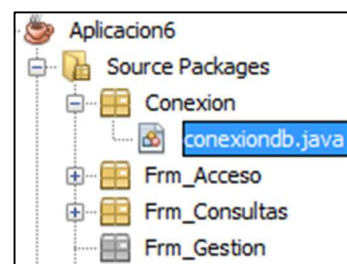


- B) Crear la base de datos "ventas" y una tabla "Usuario". Vea siguiente imagen:

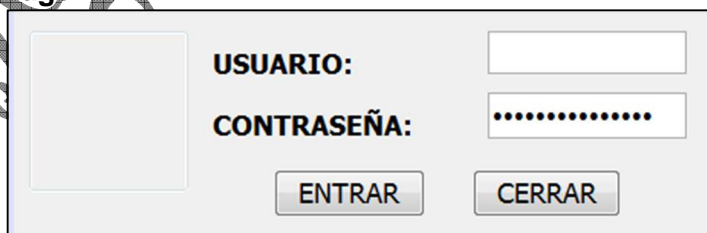
Idusuario	tipousuario	nomusuario	clave
1	ADMIN	YOLFER	123
2	USER	DIANA	123

3) DESARROLLO DEL PROYECTO.

- A) Crear paquetes, Vea la imagen.
B) En el paquete "Conexion" copiar la clase "Conexionbd" implementada en la sesión anterior.



- C) Diseñamos el formulario "frm_login", donde debemos incluir controles. Vea la siguiente imagen:



- A) Importar el Espacio de Nombres.

```
import Conexion.*;
import java.sql.*;
import javax.swing.JOptionPane;
```

- B) Programar a nivel de clase



```
public class frm_Acceso extends javax.swing.JFrame {
    Connection cone = conexiondb.conectarbd();
    Statement st = null;
    ResultSet rs = null;

    //PROCEDIMIENTO QUE FINALIZA LA APLICACIÓN
    void Salir(){
        int r=JOptionPane.showConfirmDialog(this, "?","IMPORTANTE",
        JOptionPane.YES_NO_OPTION,JOptionPane.CANCEL_OPTION);
        //Evaluar el cierre de ventana
        if (r==0) System.exit(0);
    }
}
```

C) Programar en el botón ENTRAR.

```
try {

    String user, pass, sql, Tipouser="";
    user = txtUsuario.getText(); pass = txtClave.getText();

    sql = "Select * from usuario Where nomusuario='"+user+"' && clave='"+pass+"' ";
    st = cone.createStatement();
    ResultSet rs = st.executeQuery(sql);
    while (rs.next()) {
        Tipouser = rs.getString(2);
    }

    if(Tipouser.equals("ADMIN") ){
        frm_Principal wPrincipal = new frm_Principal();
        wPrincipal.setVisible(true);
        this.setVisible(false);
    }
    else if(Tipouser.equals("USER") ){
        JOptionPane.showMessageDialog(null, "VENTANA SIN PRIVILEGIOS");
    }else if(Tipouser.equals("")){
        JOptionPane.showMessageDialog(null, "USUARIO NO VALIDO");
    }

} catch (SQLException ex) {
    System.out.println(ex);
}
```

D) Programar en el botón SALIR.

```
Salir();
```

E) Probar la Aplicación.



4

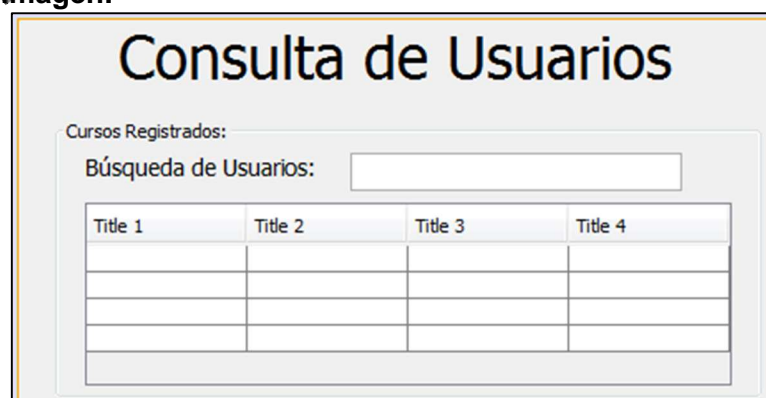
1. Diseñamos el formulario “frm_Principal”, donde debemos **incluir controles**. Vea la siguiente imagen:



2. Codificar En El Boton MATRICUAR

```
frm_carrera carr = new frm_carrera();
carr.setVisible(true);
this.setVisible(false);
```

1. Diseñamos el formulario “frm_login”, donde debemos **incluir controles**. Vea la siguiente imagen:



Title 1	Title 2	Title 3	Title 4



A) Importar el Espacio de Nombres.

```
import Conexion.*;
import java.sql.*;
import javax.swing.JOptionPane;
```

B) Programar a nivel de clase

```
public class frm_CUusuario extends javax.swing.JFrame {
    //A) Objetos Conexion, Statement, ResultSet
    Connection cone = conexiondb.conectarbd();
    Statement st = null;
    ResultSet rs = null;
    //B. Obeto ModeloTabla
    DefaultTableModel dtm = new DefaultTableModel();
    //3. PROCEDIMIENTO TRAE INFORMACIÓN
    public void TraerProveedor(){
        try {
            st = cone.createStatement();//Conexion a bd
            rs = st.executeQuery("select * from usuario;");
            String datos[] = new String[4]; //Declaración de Array
            //Limpiar Tabla
            int fila = dtm.getRowCount(); //Retornando el # Filas
            if (fila>0) //Si tiene al menos un registro
                for (int i=0;i<fila;i++)
                    dtm.removeRow(0);
            //Leer Datos
            while(rs.next()){ //Guardando datos en Vector
                datos[0] = rs.getString(1);
                datos[1] = rs.getString(2);
                datos[2] = rs.getString(3);
                datos[3] = rs.getString(4);
                dtm.addRow(datos); //Adicionando al Modelo
            }
        } catch (Exception e) {
            System.out.println(e);
        }
    }
    //2. METODO QUE TRAE LOS CURSOS SEGUN EL CODIGO O CURSO
    void GetCursoXdatos(String datos){
        try {
            //a. Definir una Consulta con Filtro (Like)
            String sql = "select * from usuario where idusuario like '"+txtcur1.getText()+"%' or nomusuario like '"+txtcur1.getText()+"%' ";
            Statement st = cone.createStatement(); //b. Crear Objeto Statement
            ResultSet rs = st.executeQuery(sql); //c. Ejecutar Statement que contiene Consulta
            ResultSetMetaData rsmd = rs.getMetaData(); //d. Crear un objeto Metadata
            int numcol = rsmd.getColumnCount(); //e. Variable que contiene Num Columnas
```




```
//f. Crear un objeto "modelo" con la finalidad de gestionar datos en la tabla.
// el objeto tiene metodos que permite administrar datos en un jTable
DefaultTableModel modelo = new DefaultTableModel();
//g. enviando el modelo al jTable
jTable1.setModel(modelo);
//h. Adicionando columnas(Campos) al jTable
for (int x=1;x<=numcol;x++){
    modelo.addColumn(rsmd.getColumnName(x));
}
//i. Adicionando registros al jTable
while(rs.next()){
    Object [] fila = new Object[numcol];
    for(int y=0 ; y<numcol;y++){
        fila [y] = rs.getObject(y+1);
    }
    modelo.addRow(fila);
}
} catch (Exception e) {
    lblmsg.setText(""+e); //k. Mensaje en caso exista un Error
}
}
```

C) Programar en el CONSTRUCTOR.

```
public frm_CUsuarios() {
    initComponents();

    //4. CONFIGURACION PLANTILLA DE TABLA
    String titulos[] ={"CÓDIGO", "USUARIO", "TIPO", "CAVE"};
    dtm.setColumnIdentifiers(titulos);
    jTable1.setModel(dtm);
    //MOSTRAR LOS ESTADIOS
    TraerProveedor();
}
```

D) Programar en el evento KEYPRESSED de txtBuscar

```
GetCursoXdatos(txtcur1.getText());
jTable1.getSelectionModel().setSelectionInterval(0,0);
```

E) Probar la Aplicación.

