**PASOS PARA CONECTAR UN FORMULARIO REALIZADO EN JAVA SE CON UNA BBDD MYSQL CON XAMPP**

1. Descargar e instalar los programas: Workbeanch – Xampp – Netbeans
2. Crear BBDD en WorkBeanch y exportarla en .sql
3. Ejecutar Xampp e importar BBDD .sql

**En Netbeans**

1. Dividir en el proyecto en paquetes siguiendo el patrón de diseño de Arquitectura ‘n’ capas (datos, lógica, presentación)
2. Importar la librería JDBC (Clase controladora que facilita la conexión a la BBDD), mysql-connector-java-5.1.49-bin.jar (Descargarla de internet).
3. En el paquete ‘lógica’ crear la clase ‘Conexión’ y dividir el código de la siguiente manera:
4. Importar las clases: java.sql.Connection - java.sql.DriverMananger - java.sql.SQLException – javax.swing.JOptionPane
5. Crear un constructor vacío: Public Conexión (){}
6. Crear variables que manejan la BBDD

Public String db = “basereserva”;

Public String url = “jdbc:mysql://127.0.0.1/” + db;

Public String db = “nombreBaseDatos”;

Public String db = “nombreBaseDatos”;

1. Crear la función:

public Connection conectar(){

Connection link = null;

Try{

// Buscar el Driver en la librería importada – Es la última

Class.forName(“org.gjt.mm.mysql.Driver”);

Link = DriverManager.getConnection(this.url,this.user,this.pass);

} catch(ClassNotFoundException | SQLException e){

JOptionPane.showConfirmDialog(null, e);

Return link;

}

1. Hacer prueba de conexión: En el paquete ‘lógica’ crear la clase ‘MDaoHabitacion’ y dividir el código de la siguiente manera:
2. Importar las clases: java.sql.Connection - java.sql.PreparedStatement - javax.swing.JOptionPane
3. Crear variables de la conexión

Private Conexión mysql = new Conexión();

Private Connection cn = mysql.conectar();

Private String sSQL = “”;

1. Crear la función de prueba:

public void insertar(){

sSQL = “INSERT INTO habitacion(numero, piso, descripcion, características, precioDiario, estado, tipoHabitacion) VALUES(?,?,?,?,?,?,?)”;

Try{

PreparedStatement pst = cn.prepareStatement(sSQL);

Pst.setString(1, “503”);

Pst.setString(2, “5”);

Pst.setString(3, “Habiación Matrimonial”);

Pst.setString(4, “Cama doble, bar”);

Pst.setDouble(5, “150000.00”);

Pst.setString(6, “desocupada”);

Pst.setString(7, “individual”);

Int n = pst.executeUpdate();

} catch(Exception e){

JOptionPane.showConfirmDialog(null, e);

Return link;

}

1. En el paquete ‘presentación’ crear una Clase ‘VHabitacion’ de la siguiente manera:

public static void main(String[] args) {

MDaoHabitacion dao = new MDaoHabitacion();

dao.insertar();

}

1. En el paquete ‘datos’ crear la clase ‘MDtoHabitacion’ con los atributos privados de la tabla, método constructor vacío y métodos setter and getter. Inicializar atributos con cualquier dato.
2. Importar a la Clase MDaoHabitación la clase anterior ‘ import datos.MDtoHabitacion‘ y modificar la función insertar() del punto ‘7-c’ de la siguiente forma:

public boolean insertar(MDtoHabitacion dto){

sSQL = "INSERT INTO habitacion(numero, piso, descripcion, caracteristicas, precioDiario, estado, tipoHabitacion) "

+ "VALUES (?,?,?,?,?,?,?)";

try {

PreparedStatement pst = cn.prepareStatement(sSQL);

pst.setString(1, dto.getNumero());

pst.setString(2, dto.getPiso());

pst.setString(3, dto.getDescripcion());

pst.setString(4, dto.getCaracteristicas());

pst.setDouble(5, dto.getPrecioDiario());

pst.setString(6, dto.getEstado());

pst.setString(7, dto.getTipoHabitacion());

int n = pst.executeUpdate();

if (n != 0) {

return true;

} else{

return false;

}

} catch (Exception e) {

JOptionPane.showConfirmDialog(null, e);

return false;

}

}

1. En la Clase ‘VHabitacion’ realizar la siguiente modificación al código de la siguiente manera:

public static void main(String[] args) {

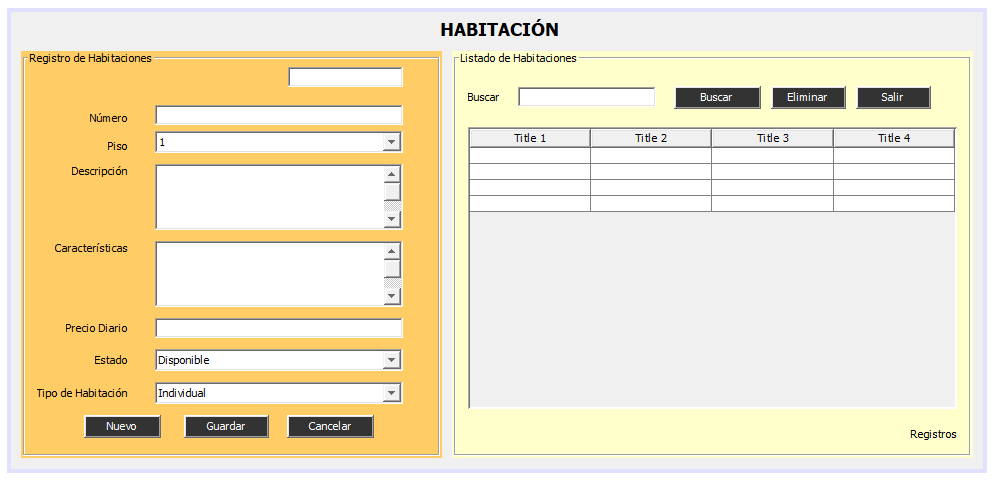
MDtoHabitacion dto = new MDtoHabitacion();

MDaoHabitacion dao = new MDtoHabitacion();

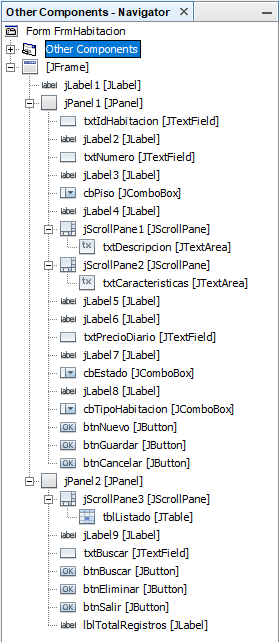
System.out.println(dao.insertar(dto));

}

1. Crear el formulario: Eliminar la clase ‘VHabitacion’ y en el mismo paquete ‘presentacion’ crear la clase JFrame Form ‘VHabitacion’
2. Dibujar el siguiente formulario:



1. Para cada uno de los controles, cambiar el valor de las variables como se indica a continuación:



1. Escribir el siguiente en la función de evento ‘btnGuardar’, para insertar los datos:

MDtoHabitacion dto = new MDtoHabitacion();

MDaoHabitacion dao = new MDaoHabitacion();

dto.setNumero(txtNumero.getText());

int sel = cbxPiso.getSelectedIndex();

dto.setPiso(cbxPiso.getItemAt(sel));

dto.setDescripcion(txtDescripcion.getText());

dto.setCaracteristicas(txtCaracteristicas.getText());

dto.setPrecioDiario(Double.parseDouble(txtPrecioDiario.getText()));

sel = cbxEstado.getSelectedIndex();

dto.setEstado(cbxEstado.getItemAt(sel));

sel = cbxTipoHabitacion.getSelectedIndex();

dto.setTipoHabitacion(cbxTipoHabitacion.getItemAt(sel));

dao.insertar(dto);

1. Crear la función mostrar() en la Clase ‘MDaoHabitacion’ y escribir el siguiente código
2. Importar las Clases: import java.sql.Statement - import java.sql.ResultSet – import javax.swing.table.DefaultTableModel;
3. En los atributos del objeto crear un acumulador para los registros: public Integer totalRegistros;
4. Crear la Función: public DefaultTableModel mostrar(String buscar){}
5. Iniciar el total de registros en 0: totalRegistros = 0;
6. Dentro de ella crear un objeto: DefaultTableModel modelo;
7. Crear un arreglo con los títulos: String[] titulos = {"ID", "Número", "Piso", "Descripción", "Características", "Precio", "Estado", "Tipo Habitación"};
8. Crear un arreglo para los registros: String[] registros = new String[8];
9. Crear el objeto modelo y en el constructor pasarle los títulos: modelo = new DefaultTableModel(null, titulos);
10. Crear la consulta SQL para llamar los datos: sSQL = "SELECT \* FROM habitacion WHERE piso LIKE '%" + buscar + "%' ORDER BY idHabitacion";
11. Crear los Try and Catch para capturar las excepciones, darle manejo a los errores y capturar los datos de la BBDD, así:

try {

Statement st = cn.createStatement();

ResultSet rs = st.executeQuery(sSQL);

while (rs.next()) {

registros[0] = rs.getString("idHabitacion");

registros[1] = rs.getString("numero");

registros[2] = rs.getString("piso");

registros[3] = rs.getString("descripcion");

registros[4] = rs.getString("caracteristicas");

registros[5] = rs.getString("precioDiario");

registros[6] = rs.getString("estado");

registros[7] = rs.getString("tipoHabitacion");

totalRegistros += totalRegistros;

modelo.addRow(registros);

}

return modelo;

} catch (Exception e) {

JOptionPane.showConfirmDialog(null, e);

return null;

}

1. Crear la función ocultarIdHabitacion() en la Clase ‘VHabitacion’ y escribir el siguiente código:

public void ocultarIdHabitacion(){

tblListado.getColumnModel().getColumn(0).setMaxWidth(0);

tblListado.getColumnModel().getColumn(0).setMinWidth(0);

tblListado.getColumnModel().getColumn(0).setPreferredWidth(0);

}

1. Crear la función mostrar() en la Clase ‘VHabitacion’ y escribir el siguiente código:

public void mostrar(String buscar){

try {

DefaultTableModel modelo;

MDaoHabitacion dao = new MDaoHabitacion();

modelo = dao.mostrar(buscar);

tblListado.setModel(modelo);

ocultarIdHabitacion();

lblRegistros.setText("Total Registros " + Integer.toString(dao.totalRegistros));

} catch (Exception e) {

JOptionPane.showConfirmDialog(rootPane, e);

}

}

**Nota:** No olvidar llamarlo en el constructor.

1. Escribir el siguiente en la función de evento ‘btnBuscar’, para insertar los datos: mostrar(txtBuscar.getText());
2. Escribir las funciones ‘inhabilitar’ y ‘habilitar’ debajo del constructor para ocultar o mostrar la funcionalidad de los controles. Al terminar llame la función ‘inhabilitar’ desde el constructor:

public VHabitacion() {

initComponents();

this.setLocationRelativeTo(null);

inhabilitar();

}

public void inhabilitar(){

txtIdHabitacion.setVisible(false);

txtNumero.setEnabled(false);

cbxPiso.setEnabled(false);

txtDescripcion.setEnabled(false);

txtCaracteristicas.setEnabled(false);

txtPrecioDiario.setEnabled(false);

cbxEstado.setEnabled(false);

cbxTipoHabitacion.setEnabled(false);

btnGuardar.setEnabled(false);

btnCancelar.setEnabled(false);

btnEliminar.setEnabled(false);

txtIdHabitacion.setText("");

txtNumero.setText("");

txtDescripcion.setText("");

txtCaracteristicas.setText("");

txtPrecioDiario.setText("");

}

public void habilitar(){

txtIdHabitacion.setVisible(false);

txtNumero.setEnabled(false);

cbxPiso.setEnabled(false);

txtDescripcion.setEnabled(false);

txtCaracteristicas.setEnabled(false);

txtPrecioDiario.setEnabled(false);

cbxEstado.setEnabled(false);

cbxTipoHabitacion.setEnabled(false);

btnGuardar.setEnabled(false);

btnCancelar.setEnabled(false);

btnEliminar.setEnabled(false);

txtIdHabitacion.setText("");

txtNumero.setText("");

txtDescripcion.setText("");

txtCaracteristicas.setText("");

txtPrecioDiario.setText("");

}