

## MYSQL

Crear una base de datos llamada “agenda” y, en ella, crear una tabla llamada “Persona” con los siguientes datos: id – apellidos – nombres – celular.

```
CREATE DATABASE agenda;
```

```
USE agenda;
```

```
CREATE TABLE Persona(  
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
apellidos NVARCHAR(50),  
nombres NVARCHAR(50),  
celular NVARCHAR(15)  
);
```

```
/*Alta*/
```

```
INSERT INTO Persona (apellidos, nombres, celular) VALUES ('Torres', 'Antonio', '11-4678-2345');
```

```
INSERT INTO Persona (apellidos, nombres, celular) VALUES ('Fuentes', 'Carlos', '11-4678-2345');
```

```
/*Consulta*/
```

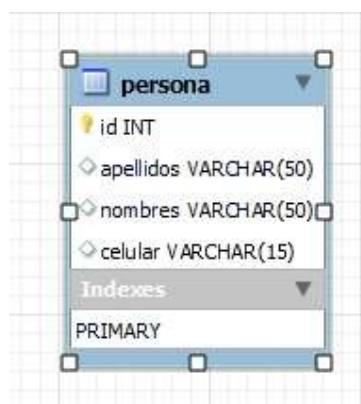
```
SELECT * FROM Persona;
```

```
/*Modificación*/
```

```
UPDATE Persona SET apellidos = 'Sierra' WHERE id = 1;
```

```
/*Baja*/
```

```
DELETE FROM Persona WHERE id=2;
```

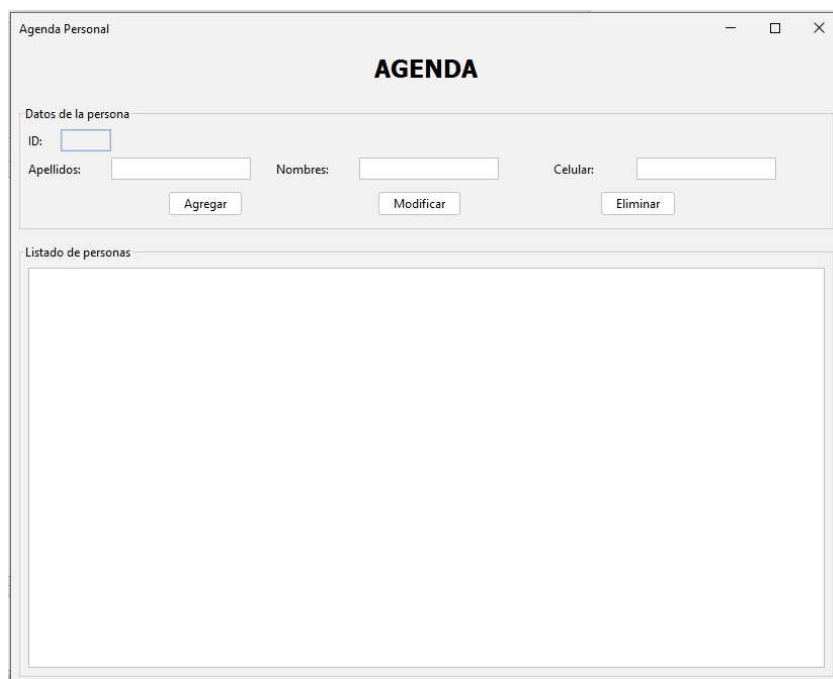


## Netbeans

Crear un nuevo Proyecto: “Agenda” con el Main Class llamado: “Principal”.

Crear un nuevo JFrame Form llamado: “Formulario”.

- Colocar el texto “Agenda Personal” en la propiedad “title”.
- Colocar 3 Panel paralelos en forma horizontal.
- En el Panel superior colocar un JLabel con el texto “AGENDA” (Tahoma - 24 – Bold).
- En el segundo Panel (el del medio) elegir como border “Titled Border” y en la propiedad del borde colocar “Datos de la persona”.
- En el mismo Panel colocar 4 Label, 4 Text Field y 3 Button.
- El texto del JLabel2 es “ID: ” y el nombre de variable es “lblId”.
- El texto del JLabel3 es “Apellidos: ” y el nombre de variable es “lblApellidos”.
- El texto del JLabel4 es “Nombres: ” y el nombre de variable es “lblNombres”.
- El texto del JLabel5 es “Teléfonos: ” y el nombre de variable es “lblCelular”.
- El texto del jTextField1 es “”, el nombre de variable es “txtId” y destildar la propiedad “editable”.
- El texto del jTextField2 es “” y el nombre de variable es “txtApellidos”.
- El texto del jTextField3 es “” y el nombre de variable es “txtNombres”.
- El texto del jTextField4 es “” y el nombre de variable es “txtCelular”.
- El texto del Button1 es “Agregar” y el nombre de variable es “btnAgregar”.
- El texto del Button2 es “Modificar” y el nombre de variable es “btnModificar”.
- El texto del Button3 es “Eliminar” y el nombre de variable es “btnEliminar”.
- En el tercer Panel (el de abajo) elegir como border “Titled Border” y en la propiedad del borde colocar “Listado de personas”.
- En el mismo Panel colocar un Table.
- En el jTable1, con el botón derecho elegir “Table Contents”, la pestaña “Columns” y borrar los 4 objetos.
- En el jTable1 colocar el nombre de variable “tbPersona”.



Descargar el conector de MySQL para Java desde la página de descarga de MySQL:

[dev.mysql.com/downloads/connector/j/](https://dev.mysql.com/downloads/connector/j/)

MySQL Community Downloads

Connector/J

General Availability (GA) Releases Archives

Connector/J 8.1.0

Select Operating System:

Platform Independent

Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-j-8.1.0.tar.gz) MD5: 4a95d62b0cfba668b92ffc62ae7ee266 | Signature

Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-j-8.1.0.zip) MD5: d745362823ec4d37fa0687746d4a1b9 | Signature

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

ORACLE © 2023 Oracle

Privacy / Do Not Sell My Info | Terms of Use | Trademark Policy | Preferencias sobre cookies

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login » using my Oracle Web account

Sign Up » for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

No thanks, just start my download.

ORACLE © 2023 Oracle

Privacy / Do Not Sell My Info | Terms of Use | Trademark Policy | Preferencias sobre cookies

Seguramente el archivo se haya descargado en la carpeta “Descargas”. Verificar que ahí esté el archivo: `mysql-connector-j.8.1.0 (.zip)`  
Hacerle doble click y extraer en el “Escritorio”.

Hacer click con el botón derecho sobre el paquete “Libraries” y seleccionar “Add JAR/Folder...”  
Buscar en el “Escritorio”, dentro de la carpeta “mysql-connect-j-8.1.0” el archivo:

`mysql-connector-j-8.1.0(.jar)`

Crear una nueva Clase llamada “Conexion”.

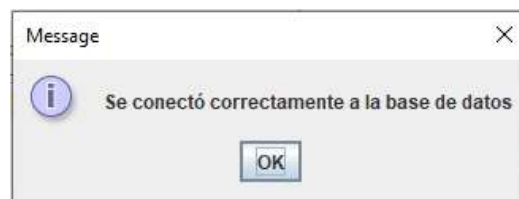
- Importamos:
  - java.sql.Connection;
  - java.sql.DriverManager;
  - javax.swing.JOptionPane;
- Crear los atributos:
  - String ip = "localhost";
  - String puerto = "3306";
  - String usuario = "root";
  - String pass = "cfp403.lujan";
  - String bd = "agenda";
  - String cadena = "jdbc:mysql://" + ip + ":" + puerto + "/" + bd;
  - Connection con = null;
- Crear el método “conectar” que retorna una “Connection”:

```
public Connection conectar() {  
    try {  
        Class.forName("com.mysql.cj.jdbc.Driver");  
        con = DriverManager.getConnection(cadena,usuario,pass);  
        JOptionPane.showMessageDialog(null,"Se conectó correctamente a la base de datos");  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(null,"No se conectó a la base de datos, error: " + e.getMessage());  
    }  
    return con;  
}
```

En la clase “Principal”, para comprobar la conexión con MySQL escribir:

```
Conexion cn = new Conexion();  
cn.conectar();
```

deberá aparecer el siguiente cuadro de diálogo:



Si todo funciona, borrar las dos líneas anteriores de la clase “Principal” o colocarles delante “//” para que queden como comentarios.

Crear una nueva Clase llamada “Persona”.

- Crear los atributos:
  - int id;
  - String apellidos;
  - String nombres;
  - String celular;

Crear el constructor vacío y los Getters and Setters.

En la clase “Persona” importar:

- java.sql.CallableStatement;
- javax.swing.JOptionPane;
- javax.swing.JTextField;

Crear en la clase “Persona” el método “agregarPersona” (que no retorna nada) con tres parámetros tipo JTextField (apellidos, nombres y celular):

```
public void agregarPersona(JTextField apellidos, JTextField nombres, JTextField celular){
    this.setApellidos(apellidos.getText());
    this.setNombres(nombres.getText());
    this.setCelular(celular.getText());
    Conexion con = new Conexion();
    String comando = "INSERT INTO Persona(apellidos,nombres,celular) VALUES (?, ?, ?);";
    try {
        CallableStatement cs = con.conectar().prepareCall(comando);
        cs.setString(1, getApellidos());
        cs.setString(2, getNombres());
        cs.setString(3, getCelular());
        cs.execute();
        JOptionPane.showMessageDialog(null, "Se agregó correctamente la persona");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "No se agregó correctamente la persona, error: " +
            e.toString());
    }
}
```

En la clase “Persona” importar:

- javax.swing.JTable;
- javax.swing.table.DefaultTableModel;
- java.sql.Statement;
- java.sql.ResultSet;

Crear en la clase “Persona” el método “mostrarPersona” (que no retorna nada) con un parámetro tipo JTable (totalPersona):

```
public void mostrarPersona(JTable totalPersona){
    Conexion con = new Conexion();
    DefaultTableModel tabla = new DefaultTableModel();
    tabla.addColumn("Id");
    tabla.addColumn("Apellidos");
    tabla.addColumn("Nombres");
    tabla.addColumn("Celular");
    totalPersona.setModel(tabla);
    DefaultTableModel tabla = new DefaultTableModel();
    String sql = "SELECT * FROM Persona;";
    String[] datos = new String[4];
    Statement st;
    try {
```

```
st = con.conectar().createStatement();
ResultSet rs = st.executeQuery(sql);
while(rs.next()){
    datos[0] = rs.getString(1);
    datos[1] = rs.getString(2);
    datos[2] = rs.getString(3);
    datos[3] = rs.getString(4);
    tabla.addRow(datos);
}
totalPersona.setModel(tabla);
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, "No se pudo mostrar los registros, error: " + e.toString());
}
}
```

En la clase “Formulario” hacer doble click sobre el botón de “Agregar” y colocar en el evento:

```
Persona persona = new Persona();
persona.agregarPersona(txtApellidos, txtNombres, txtCelular);
persona.mostrarPersona(tbPersonas);
```

En la clase “Formulario” hacer click en la solapa “Source” y colocar debajo de “initComponents();” nuevamente el mismo texto y la “localización”:

```
Persona persona = new Persona();
persona.mostrarPersona(tbPersonas);
setLocationRelativeTo(null);
```

Crear en la clase “Persona” el método “seleccionarPersona” (que no retorna nada) con cinco parámetros, uno tipo JTable y 4 tipo JTextField (totalPersona, id, apellidos, nombres y celular):

```
public void seleccionarPersona(JTable totalPersona, JTextField id, JTextField apellidos, JTextField nombres,
JTextField celular){
    try {
        int fila = totalPersona.getSelectedRow();
        if (fila >= 0) {
            id.setText(totalPersona.getValueAt(fila, 0).toString());
            apellidos.setText(totalPersona.getValueAt(fila, 1).toString());
            nombres.setText(totalPersona.getValueAt(fila, 2).toString());
            celular.setText(totalPersona.getValueAt(fila, 3).toString());
        }else{
            JOptionPane.showMessageDialog(null, "Fila no seleccionada");
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error de selección, error: " + e.toString());
    }
}
```



En la clase “Formulario”, sobre la tabla, hacer click en el botón derecho y seleccionar: “Events” > “Mouse” > “mouseClicked” y colocar en el evento:

```
Persona persona = new Persona();
persona.seleccionarPersona(tbPersonas,txtId, txtApellidos, txtNombres, txtCelular);
```

Crear en la clase “Persona” el método “modificarPersona” (que no retorna nada) con cuatro parámetros tipo JTextField (id, apellidos, nombres y celular):

```
public void modificarPersona(JTextField id, JTextField apellidos, JTextField nombres, JTextField celular){
    this.setId(Integer.parseInt(id.getText()));
    this.setApellidos(apellidos.getText());
    this.setNombres(nombres.getText());
    this.setCelular(celular.getText());
    Conexion con = new Conexion();
    String comando ="UPDATE Persona SET apellidos = ?,nombres = ?,celular = ? WHERE id=?;";
    try {
        CallableStatement cs = con.conectar().prepareCall(comando);
        cs.setString(1, getApellidos());
        cs.setString(2, getNombres());
        cs.setString(3, getCelular());
        cs.setInt(4, getId());
        cs.execute();
        JOptionPane.showMessageDialog(null, "Se modificó correctamente la persona");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "No se modificó la persona, error: " + e.toString());
    }
}
```

En la clase “Formulario” hacer doble click sobre el botón de “Modificar” y colocar en el evento:

```
Persona persona = new Persona();
persona.modificarPersona(txtId,txtApellidos,txtNombres, txtCelular);
persona.mostrarPersona(tbPersonas);
```

Crear en la clase “Persona” el método “eliminarPersona” (que no retorna nada) con un parámetro tipo JTextField (id):

```
public void eliminarPersona(JTextField id){
    this.setId(Integer.parseInt(id.getText()));
    Conexion con = new Conexion();

    String comando ="DELETE FROM Persona WHERE id=?;";

    try {
        CallableStatement cs = con.conectar().prepareCall(comando);
        cs.setInt(1, getId());
        cs.execute();
        JOptionPane.showMessageDialog(null, "Se eliminó correctamente a la persona");
    }
}
```



```
} catch (Exception e) {  
    JOptionPane.showMessageDialog(null, "No se eliminó a la persona, error: " + e.toString());  
}  
}
```

En la clase “Formulario” hacer doble click sobre el botón de “Eliminar” y colocar en el evento:

```
Persona persona = new Persona();  
persona.eliminarPersona(txtId);  
persona.mostrarPersona(tbPersonas);
```