

## ACCESO A BBDD - EJEMPLO BÁSICO

Vamos a acceder a una base de datos relacional MariaDB (también valdría para una del tipo MySQL). En el Aula Virtual puedes encontrar los archivos necesarios (script y driver de conexión) junto con el código de ejemplo.

Los pasos generales para conseguir nuestro objetivo son los siguientes:

1. Instalar el sistema gestor de base de datos, es decir, instalar MariaDB
2. Cargar el script SQL
3. Conectar NetBeans con la base de datos
4. Programar la conexión a la base de datos

### 1. Instalar el sistema gestor de base de datos, es decir, instalar MariaDB

Necesitamos tener instalado MariaDB o MySQL en local, para hacer pruebas en local. La forma más sencilla es usar el paquete XAMPP, que instala Apache, PHP, **MariaDB**, PHPMyAdmin, etc. Podríamos haber instalado únicamente MariaDB, pero por comodidad, instalaremos el paquete completo.

Una vez instalado, solo necesitamos los parámetros de conexión, que por defecto son:

- IP: 127.0.0.1
- usuario: root
- contraseña: *sin contraseña*
- puerto: 3306

NOTA: también sería posible hacer pruebas conectándonos a una base de datos remota. Solo necesitamos que nos proporcionen los cuatro parámetros anteriores.

### 2. Cargar el script SQL

Este script sirve para crear la base de datos (CREATE DATABASE ...) y para crear las tablas junto con sus datos, todo junto en uno.

Si el script no llevara la creación de la base de datos, deberíamos crearla nosotros "a mano" y posteriormente cargar el script con las tablas.

Necesitamos usar algún administrador de MariaDB, como puede ser el PHPMyAdmin.

En nuestro caso, **vamos a crearnos la base de datos PERSONAS**

### 3. Conectar NetBeans con la base de datos

NetBeans lleva un pequeño gestor de la base de datos. Es útil pues nos permite lanzar y para el servicio desde el propio NetBeans, y no tenemos que usar el panel de control de XAMPP (o de MariaDB si la hemos instalado en solitario).

Necesitamos el driver o conector apropiado para nuestra base de datos. Este conector lo puedes encontrar en internet. En nuestro caso, usaremos:

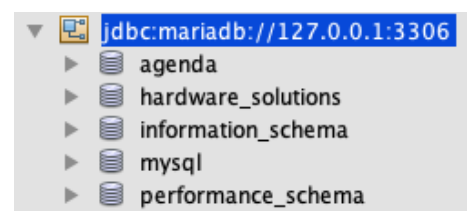
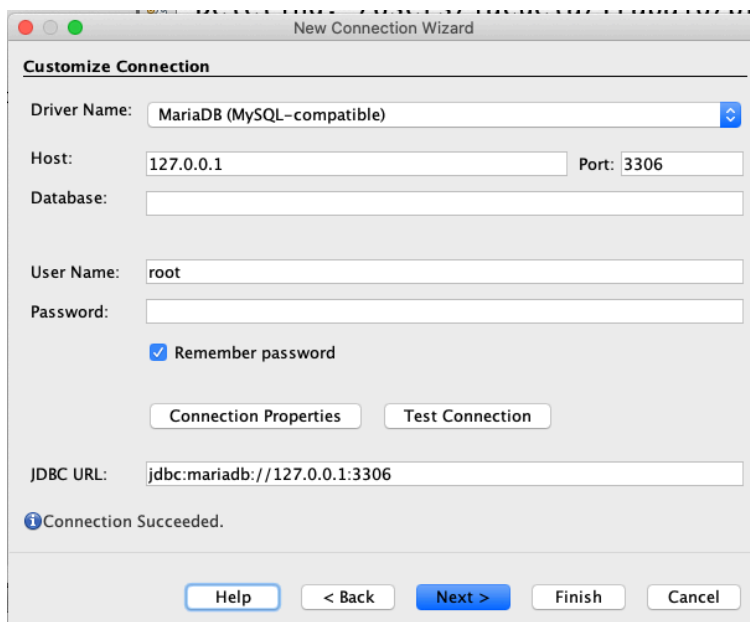
**mariadb-java-client-2.5.4.jar**

Abriremos NetBeans, y en la pestaña de servicios, instalaremos el driver. Una vez instalado, crearemos una nueva conexión usando el driver de MariaDB recién instalado.



Tendremos que indicar los parámetros de la conexión del paso 1. Si dejamos el campo *Database* vacío, podremos ver todas las bases de datos. También podemos crear una conexión a una base de datos concreta (en la conexión solo aparecerá dicha base de datos y el resto no nos molestarán).

Pulsa en Test Connection para probar la conexión antes de terminar.



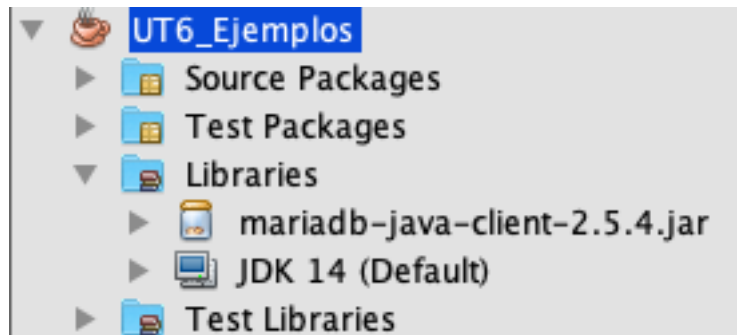
**CONEXIÓN Y SUS BBDD**

Podemos crearnos todas las conexiones que queramos.

#### 4. Programar la conexión a la base de datos

Ya solo nos queda crear nuestro programa JAVA y conectarnos a nuestra base de datos PERSONAS.

IMPORTANTE: Antes de nada, necesitamos volver a incluir el driver en nuestro proyecto. Lo haremos al igual que cuando añadimos un librería, añadiendo un archivo .jar



El código de ejemplo sería el siguiente:

```
import java.sql.*;

/**
 *
 * @author jacuela
 */

public class EjemploAccesoBD {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here

        /*******
        //      CARGAMOS EL DRIVER
        /*******
        try {
            Class.forName("org.mariadb.jdbc.Driver").newInstance();
            System.out.println("Driver <org.mariadb.jdbc.Driver> cargado.");
        } catch (Exception ex) {
            System.out.println("Error, no se ha podido cargar MariaDB JDBC
Driver");
        }

        //      try {
        //          Class.forName("com.mysql.cj.jdbc.Driver").newInstance();
        //          System.out.println("Driver <com.mysql.jdbc.Driver> cargado.");
```

```

//      } catch (Exception ex) {
//          System.out.println("Error, no se ha podido cargar el driver
<com.mysql.jdbc.Driver>.");
//      }
//

    try{

        //*****
        //    NOS CONECTAMOS A LA BASE DE DATOS PERSONAS
        //*****

        //String url = "jdbc:mysql://127.0.0.1/personas";
        String url = "jdbc:mariadb://127.0.0.1/personas";
        String username = "root";
        String password = "";

        Connection connection = DriverManager.getConnection(url, username,
password);
        Statement statement = connection.createStatement();

        //*****
        //    HACEMOS UN INSERT
        //*****

        //      statement.executeUpdate("INSERT INTO personas VALUES
(NULL,'David','Costa',19)");

        //      String unnombre="Martita";
        //      String unapellido="Gris";
        //      int unaedad=20;
        //
        //      String sql=String.format("INSERT INTO personas VALUES
(NULL,'%s','%s',%d)",unnombre,unapellido,unaedad);
        //      statement.executeUpdate(sql);
        //
        //      //statement.executeUpdate("INSERT INTO personas VALUES
(NULL,""+unnombre+"','"+unapellido+"','"+unaedad+"")");

        //      String nombres[]={ "Alfredo1","Alfredo2"};
        //      String apellidos[]={ "Perez","Olmos"};
        //      String edades[]={ "19","56"};
        //
        //      //se insertan datos en la tabla
        //      for (int i = 0; i < nombres.length; i++) {
        //          statement.executeUpdate("INSERT INTO personas VALUES (NULL, '"
+ nombres[i] + "','"+ apellidos[i] + "','"+ edades[i] + "')");
        //      }
    }

```

```
//*****  
//      HACEMOS UN SELECT  
//*****  
  
ResultSet rs = statement.executeQuery("SELECT * FROM personas");  
//ResultSet rs = statement.executeQuery("SELECT nombre,apellidos  
FROM personas");  
  
while (rs.next()) {  
  
    int id = rs.getInt("id");  
    String nombre = rs.getString("nombre");  
    String apellido = rs.getString("apellidos");  
    int edad = rs.getInt("edad");  
  
    System.out.printf("%d, %s %s, %d\n", id, nombre, apellido,  
edad);  
  
    }  
  
    rs.close();  
    statement.close();  
    connection.close();  
  
} catch (SQLException ex) {  
    System.out.println(ex);  
}  
  
}  
  
}
```