# MySQL y Java

José Jarones

## 1º Crear el proyecto con Java with ANT

Para evitar problemas al importar la libreria .jar

2º Descargar:

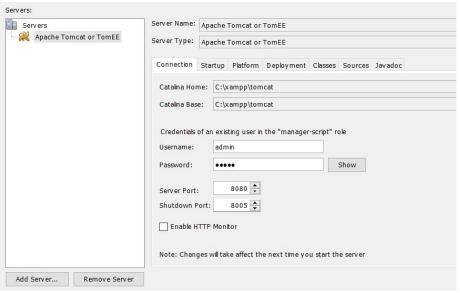
https://es.osdn.net/projects/sfnet\_id2d/downloads/jdbc%20drivers/mysql-connector-java-5.1.15-bin.jar/

3° En libraries, con el botón derecho importar.

## Crear base de datos mysql con phpmyadmin

En xampp activar apache y mysql para crear base de datos.

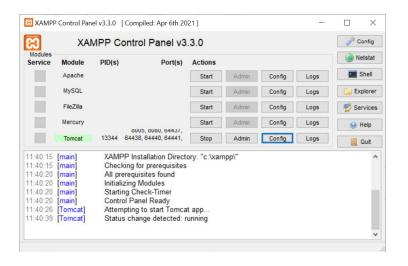
Configurar servidor tomcat en netbeans si es necesario en tools > servers



Tendrás que añadir un usuario a la configuración de Tomcat si no lo tienes creado anteriormente.

## Añadimos el usuario tomcat para administrar tomcat

Pulsamos sobre config tomcat-users.xml



```
Archivo Edición Formato Ver Avuda
<?xml version="1.0" encoding="UTF-8"?>
 Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements. See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License. You may obtain a copy of the License at
     http://www.apache.org/licenses/LICENSE-2.0
  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied
  See the License for the specific language governing permissions and
 limitations under the License
ctomcat-users xmlns="http://tomcat.apache.org/xml"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
 By default, no user is included in the "manager-gui" role required
 to operate the "/manager/html" web application. If you wish to use this app
  you must define such a user - the username and password are arbitrary.
  Built-in Tomcat manager roles:
   - manager-gui - allows access to the HTML GUI and the status pages
   - manager-script - allows access to the HTTP API and the status pages
   - manager-imx - allows access to the DMX proxy and the status pages
   - manager-status - allows access to the status pages only
  The users below are wrapped in a comment and are therefore ignored. If you
  wish to configure one or more of these users for use with the manager web
  application, do not forget to remove the < | . . . . > that surrounds them. You
 will also need to set the passwords to something appropriate.
 cuser username="admin" password="cmust-be-changed>" roles="manager-gui"/>
 <user username="robot" password="<must-be-changed>" roles="manager-script"/>
 The sample user and role entries below are intended for use with the
 examples web application. They are wrapped in a comment and thus are ignored
 when reading this file. If you wish to configure these users for use with the
  examples web application, do not forget to remove the <1....> that surrounds
 them. You will also need to set the passwords to something appropriate.
 <role rolename="tomcat"/>
 <role rolename="role1"/>
 <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
 cuser username="both" password="cmust-be-changed>" roles="tomcat.role1"/>
 <user username="role1" password="<must-be-changed>" roles="role1"/>
```

```
<role rolename="manager-gui"/>
<role rolename="admin-gui"/>
<user username="tomcat" password="pass" roles="admin-gui,manager-gui"/>
```

#### Comandos

Connection (Realizar la conexión de la bd)

Statement (Permite ejecutar sentencias -SQL)

ResultSet (Variable que permite almacenar la info de una consulta)

Driver:

Class.forName("com.mysql.jdbc.Driver")

Cadena de conexión:

DriverManager.getConnection("jdbc:mysql://localhost/bd?user=root&password=")

#### Necesito crear base de datos

Ejemplo: Base de datos alumnos creada en phpMyAdmin (Utilizamos XAMP)



## En nuevo proyecto Inicializo variables anteriores

Importar ="java.sql.\*"

```
Connection cnx = null;
Statement sta= null;
ResultSet rs= null;
%>
```

Class.forName("com.mysql.jdbc.Driver") //Utilizo el driver dentro de un bloque try catch

```
try{
    Class.forName("com.mysql.jdbc.Driver");
    cnx= DriverManager.getConnection("jdbc:mysql://localhost/nueva?user=root&password=");
}catch(Exception e) {}
```

# Ejecutar sentencia y recoger resultado

}catch(Exception e) {}

```
Connection cnx = null;
Statement sta= null;
ResultSet rs= null;
try{
    Class.forName("com.mysql.jdbc.Driver");
    cnx= DriverManager.getConnection("jdbc:mysgl://localhost/reportes?user=root&password=");
    sta=cnx.createStatement();
    rs=sta.executeQuery("select * from alumnos");
    System.out.print("");
    while (rs.next()) {
         System.out.print(rs.getString(1)); //Saco columna 1
         System.out.print(rs.getString(2)); //Saco columna 2
         System.out.print(rs.getString(3));
         System.out.print(rs.getString(4));
         //Añadimos una tercera columna con un enlace a eliminar este usuario
      System.out.println("\n");
```

## Resultado

```
run:
1Jose2° DAM18

2Ana2° DAM20

BUILD SUCCESSFUL (total time: 0 seconds)
```

#### Tarea

En la base de datos anterior, crea una tabla llamada Asignaturas que contenga:

Nombre (Ej. Despliegue de aplicaciones multiplataforma)

Siglas (DAM)

Curso(2° de DAM)

Guarda todas las asignaturas de 2º de DAM en la tabla.

Crea un nuevo proyecto donde se conecte a la base de datos y muestre dentro de un jtable todas las asignaturas que hay en la base de datos.

## Insertar. Crear un formulario

Creamos un formulario para recoger los datos a insertar, y un botón insertar para ejecutar la acción insertar.

#### Insertar

Recoger los parámetros del formulario (comprobar si se ha enviado bosino da error)

```
<8
Connection cnx = null;
Statement sta= null:
ResultSet rs= null;
        if (request.getParameter ("Enviar") !=null) {
 try{
     Class.forName("com.mysql.jdbc.Driver");
     cnx= DriverManager.getConnection("jdbc:mysql://localhost/nueva?user=root&password=");
     sta=cnx.createStatement();
     rs=sta.executeQuery("select * from alumnos");
     String nombre= request.getParameter("nombre");
     String curso = request.getParameter("curso");
     String sentencia="insert into alumnos values('"+nombre+"','"+curso+"')";
     sta.executeUpdate(sentencia);
 }catch(Exception e) {}
```

### Eliminar

Debemos seleccionar el valor a eliminar. Cogiendo el id correspondiente.

#### eliminar

```
try{
   Class.forName("com.mysql.jdbc.Driver");
   cnx= DriverManager.getConnection("jdbc:mysql://localhost/nueva?user=root&password=");
    sta=cnx.createStatement();
   String codigo = request.getParameter("cod");
   String sentencia="delete from alumnos where Nombre='"+codigo+"'";
   out.print(sentencia);
    sta.executeUpdate (sentencia);
   request.getRequestDispatcher("index.jsp").forward(request, response);
}catch(Exception e) {}
```

#### Tarea

Crea un nuevo proyecto y una nueva base de datos.

Debes crear la ventana principal(donde se mostrarán los datos y con el botón eliminar(para eliminar datos) y una ventana secundaria donde se añadirán nuevos datos.

La base de datos será de libros y tendrá los siguientes campos:

idLibro (int)

Título (varchar)

Autor (varchar)

Imagen(varchar) En imagen añadiremos la dirección de la imagen de la portada del libro.