



# AUT04\_02 Consumo Eléctrico con JSP

## 1. Objetivos

Los objetivos de esta actividad son:

- Reemplazar el servlet por una JSP (realmente es un servlet también) a la hora de generar la vista

## 2. Crear aplicación ConsumoElectricoJsp\_nombre

Se creará una aplicación Maven de tipo Web usando Java Enterprise Edition 7 y Glassfish 4.1.x.

### 2.1 Crear formulario de búsqueda HTML

Será una página **index.html** con un formulario de búsqueda. En este caso con método **GET**.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Buscador</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    <h1>Buscador de clientes</h1>
    <form action="/ConsumoElectricoJsp_nombre/buscadorClientes" method="GET">

      <fieldset>
        <legend>Datos cliente</legend>
        <label for="txtNombre">Nombre</label>
        <input id="txtNombre" type="text" name="nameToSearch"/>
        <input type="submit" value="Enviar"/>
      </fieldset>

    </form>
  </body>
</html>
```

### 2.2 Fichero .properties de configuración de la conexión a base de datos

Se creará un fichero **consumoelectrico.properties** que se ubicará en la carpeta raíz de “source packages” en resources. Su contenido será el siguiente:

```
database.driver=com.mysql.jdbc.Driver
database.url=jdbc:mysql://localhost:3306/consumoelectrico?profileSQL=true
```

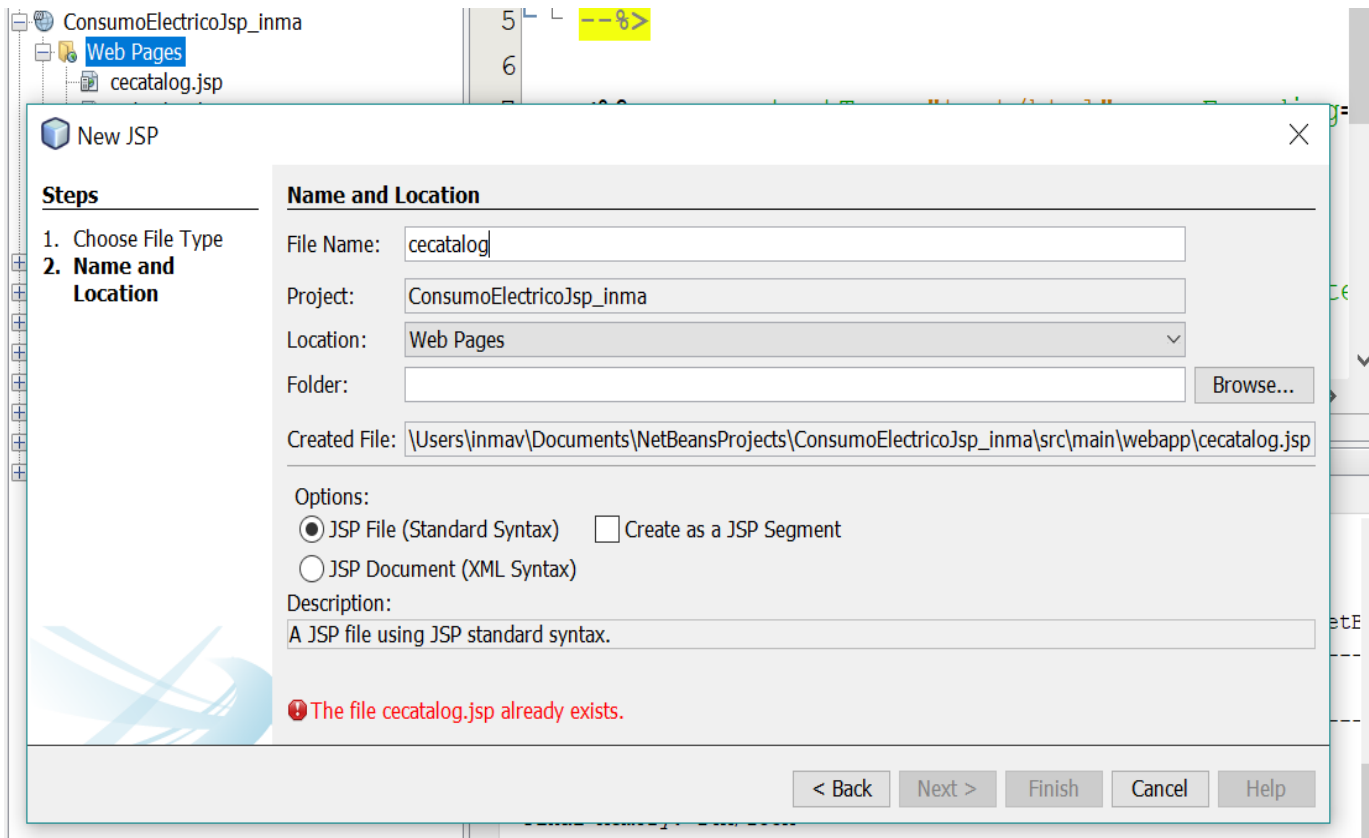


```
database.user=2dawa  
database.password=2dawa  
database.pageSize=10
```

- database.driver. Nombre cualificado de la clase Java del conector a la base de datos.
- database.url. Cadena de conexión a la base de datos.
- database.user. Usuario de la base de datos.
- database.password. Contraseña del usuario de la base de datos.
- database.pageSize. Número máximo de registros por consulta.

## 2.3 Web component de tipo JSP que realizará las búsquedas de clientes

Crearemos una JSP usando la sintaxis JSP estándar (no XML).



La ubicaremos en la raíz de **Web Pages** y de nombre **cecatalog** (se creará cecatalog.jsp)

## 2.4 Código de la JSP

La JSP realizará lo siguiente

1. Recogerá el valor del campo nameToSearch,.
2. Obtendrá una conexión del DriverManager, utilizando la cadena de conexión JDBC, así como el nombre de usuario y contraseña de consumoelectrico.properties. En este caso el nombre de fichero de propiedades sabemos que siempre es **consumoelectrico**.



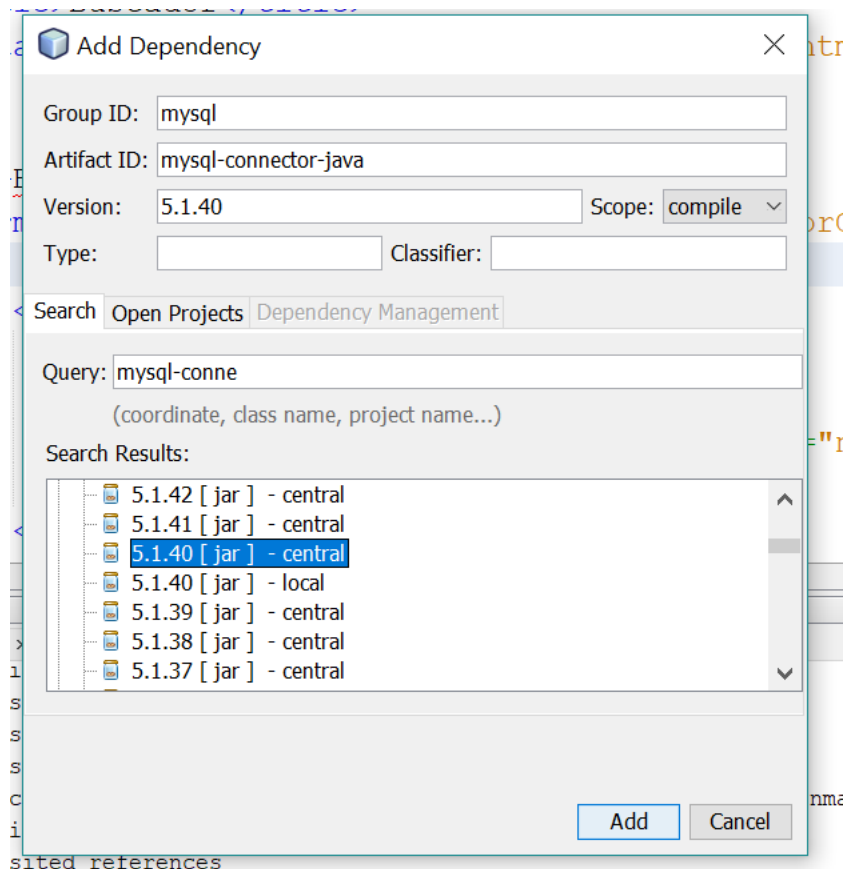
3. A partir de la conexión creará un objeto Statement.
4. Se ejecutará el statement con la siguiente consulta SQL: “SELECT \* FROM misclientes LIMIT 0,20”, asignándose el resultado a un objeto de tipo ResultSet.

**NOTA: LIMIT** es una cláusula propia de MySQL que nos permite paginar. El primer valor es el offset del primer elemento y el segundo el tamaño de página. Ejemplo: LIMIT 0, 20 => 20 registros comenzando por el primero. LIMIT 1, 20 => 20 registros comenzando por el 21.

5. Se generará una tabla HTML de respuesta.
  1. La fila de cabecera tendrá los títulos N.º, Nombre Apellido, Ciudad, Provincia de cada cliente.
  2. Las filas de datos se generarán recorriendo el objeto ResultSet obtenido en el paso 4. Se generará una fila por cliente, con celdas con los valores de los campos nombre y apellidos juntas, 2+3, significa que concatenaremos los valores de la columna 2 y 3 (con un espacio entre ambas).
6. Se cerrarán objetos de base de datos.

**NOTA:** Se utilizará una hoja de estilos CSS.

## 2.5 Añadir dependencias de terceros. Debería funcionar también con la 8.xx ó 6.xx



Añadir el driver de MySQL.



## 2.6 Ejemplo de petición y respuesta

### 2.6.1 Página de búsqueda

## Buscador de clientes

Datos cliente

Nombre

Enviar

### 2.6.2 Página de resultados

## Cientes de consumo eléctrico

Nombre	Poblacion	Provincia
Abel Pi??n	Ferrol	Coru?a, La
Abelardo Pintor	Ferrol	Coru?a, La
Ada Pintos	Ferrol	Coru?a, La
Adri?n Pita	Ferrol	Coru?a, La
Adriana Piteira	Ferrol	Coru?a, La
?gata Poceiro	Ferrol	Coru?a, La
?gueda Pol	Ferrol	Coru?a, La
Aida Polveiro	Ferrol	Coru?a, La
Aina Pombo	Ferrol	Coru?a, La
Ainhua Po?e	Ferrol	Coru?a, La