



## UNIVERSIDAD DON BOSCO FACULTAD DE ESTUDIOS TECNOLÓGICOS

**CICLO: 01/ 2013**

### ***GUIA DE LABORATORIO #7***

**Nombre de la** SERVLETS Y JDBC

**Practica:**

**Lugar de** CENTRO DE COMPUTO

**Ejecución:**

**Tiempo Estimado:** 3 HORAS CLASE

**MATERIA:** PROGRAMACIÓN ORIENTADA A OBJETOS I

**DOCENTES:** MAURICIO GONZÁLEZ

### **I. OBJETIVOS.**

Que el estudiante

- Aprenda a utilizar los sevlets de Java.
- Combinar los servlets con JDBC.

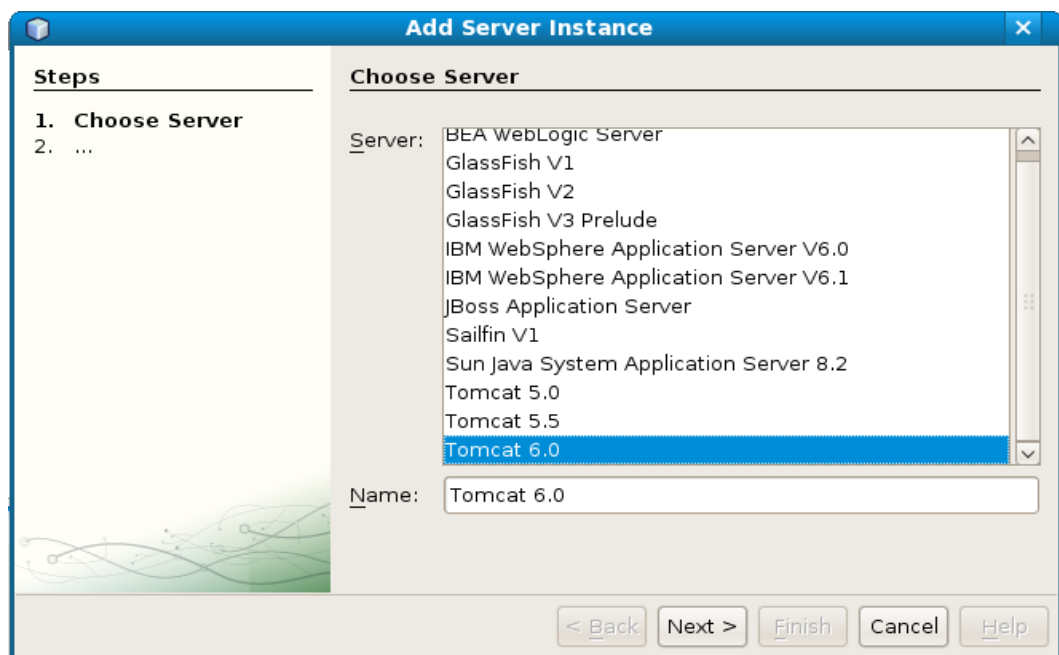
### **II. INTRODUCCIÓN.**

#### **Creación de un Servlet con Netbeans**

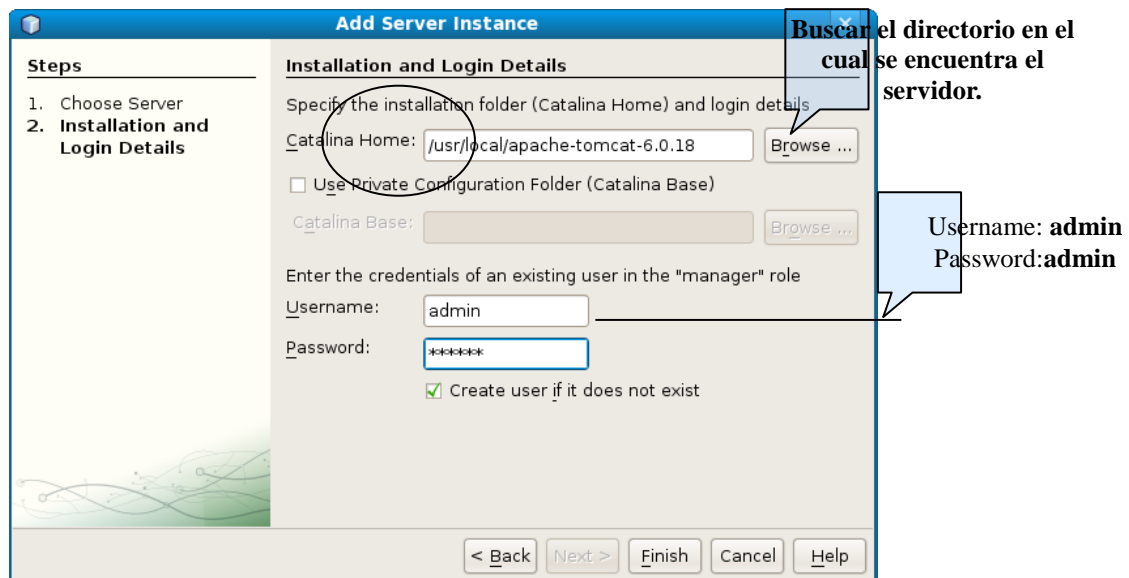
##### **Paso 1**

**Para esta parte necesitamos agregar un servidor el cual utilizaremos para nuestras aplicaciones, para ello seguir los siguiente pasos:**

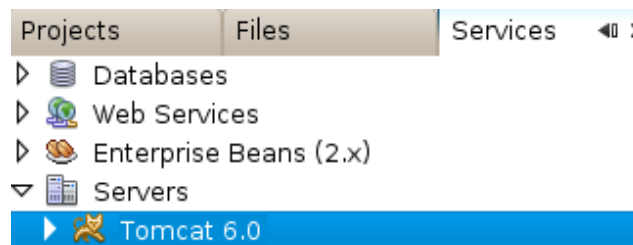
Hacer clic en la pestaña **Services** , dar click derecho sobre la opción **Servers**, seleccionar el servidor a utilizar, para esta guía utilizaremos Tomcat 6.0, dar click en **Next**.



Aparecerá una pantalla como la siguiente



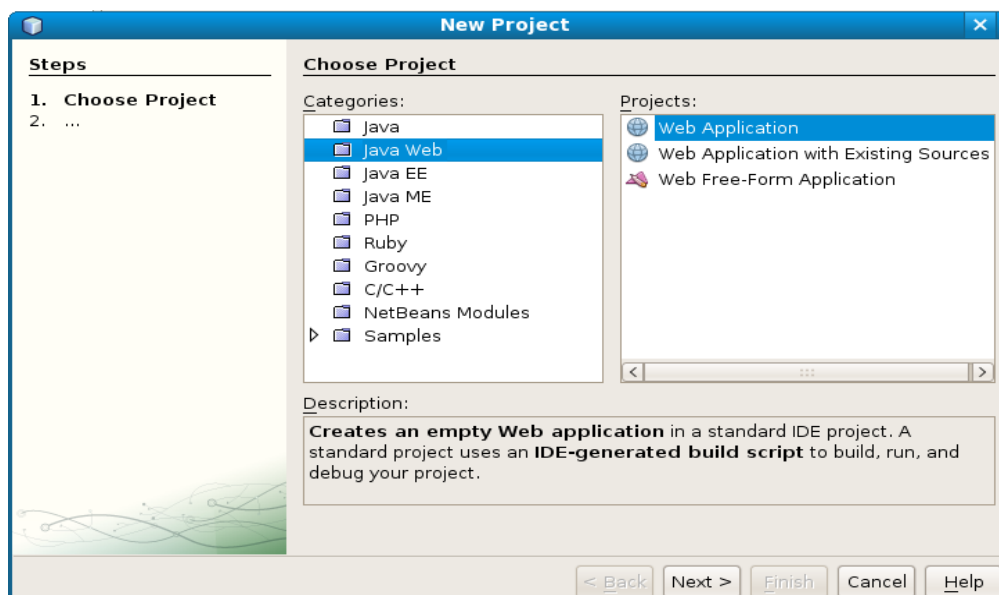
Luego de haber ingresado todo lo necesario se habilitara el botón de **Finish**, dar click en el para que el servidor a utilizar quede agregado, así como lo muestra la siguiente figura.



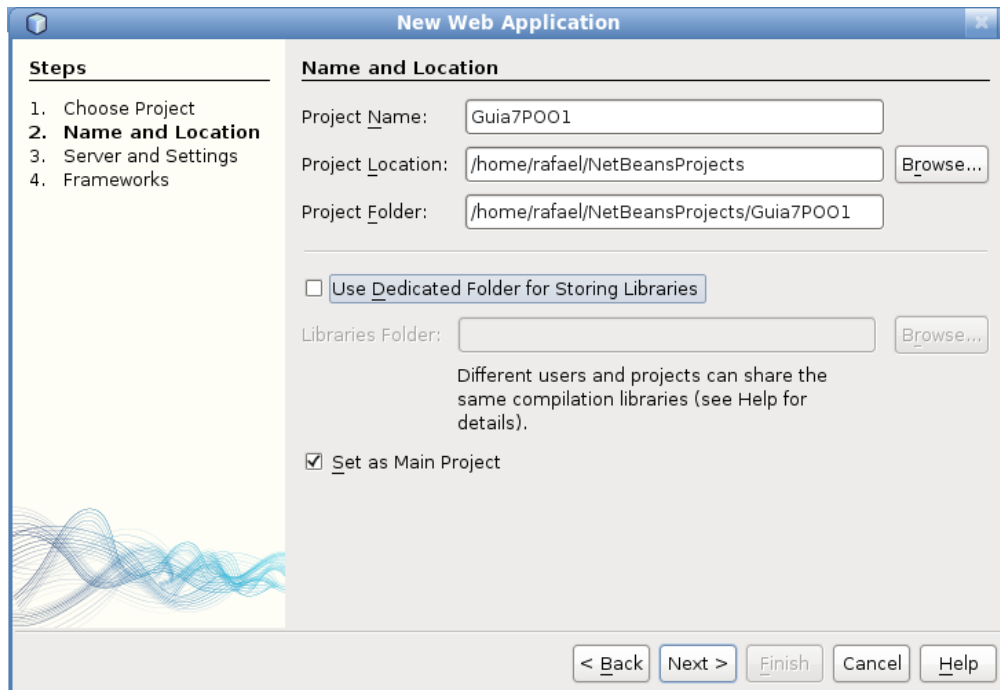
## Paso 2

Después de haber agregado el servidor, ahora crearemos nuestra proyecto web, en el cual crearemos nuestros servlets.

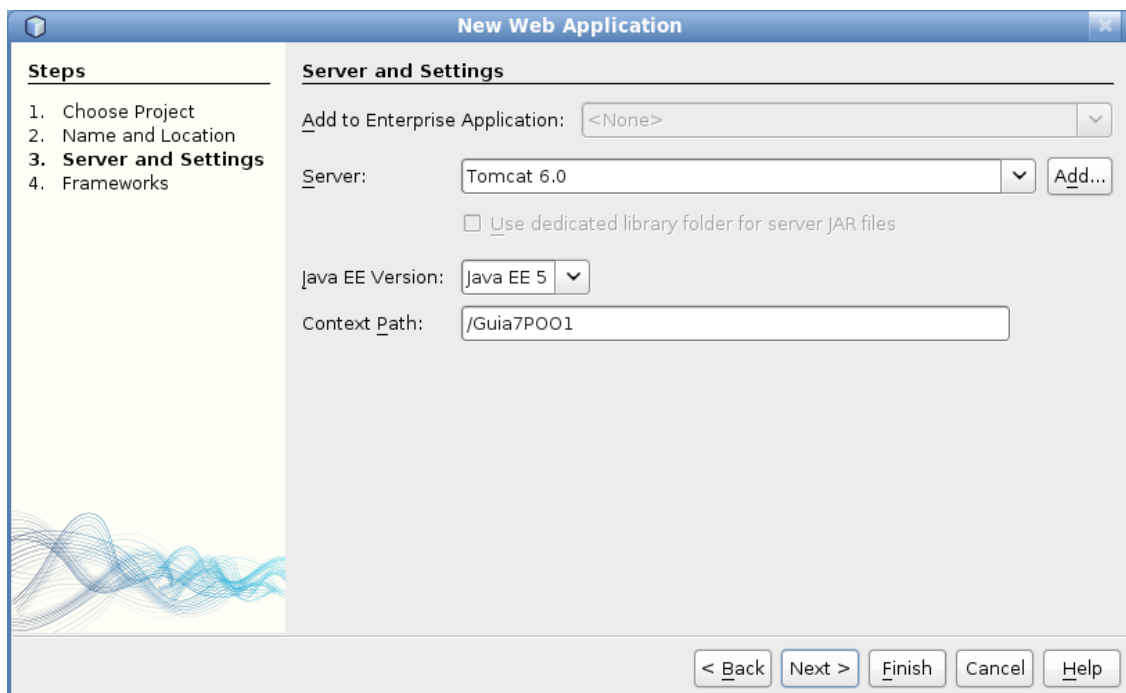
1. Iniciamos NetBeans y seleccionamos File \ New Project...
2. Se abre un diálogo que nos solicita el tipo de proyecto, seleccionamos Web Application, ver la siguiente ventana y luego dar click en **Next**.



3. Agregar al nombre del proyecto **Guia7POO1**, dar click en **Next**.



4. El siguiente paso, nos solicita el servidor web (contenedor de Servlets a utilizar). Del combo Server, seleccionamos Tomcat 6.0.



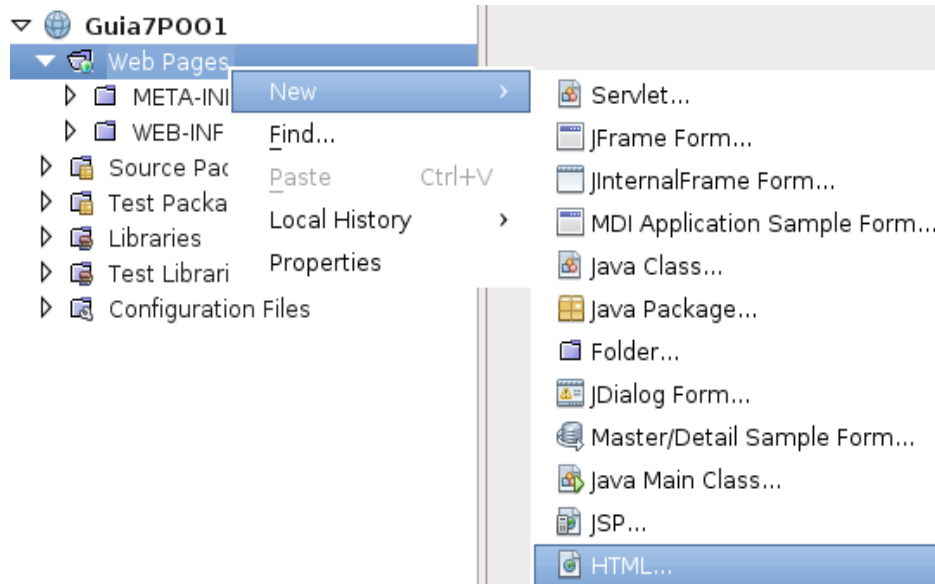
5. Finalmente, nos permite seleccionar el o los frameworks a utilizar (Spring, Struts, JSF, etc). No seleccionamos nada, por ahora, para este ejemplo, no es necesario.

NetBeans crea por su cuenta el proyecto, una estructura de directorios, y dentro de la carpeta Web Pages un archivo index.jsp, que será el punto de partida de nuestra aplicación. Si bien es de extensión JSP, por ahora no escribiremos código JSP, sino simplemente un formulario HTML. En este formulario HTML definiremos en el atributo action el nombre del servlet que se ejecutará al enviar (submit) el formulario.

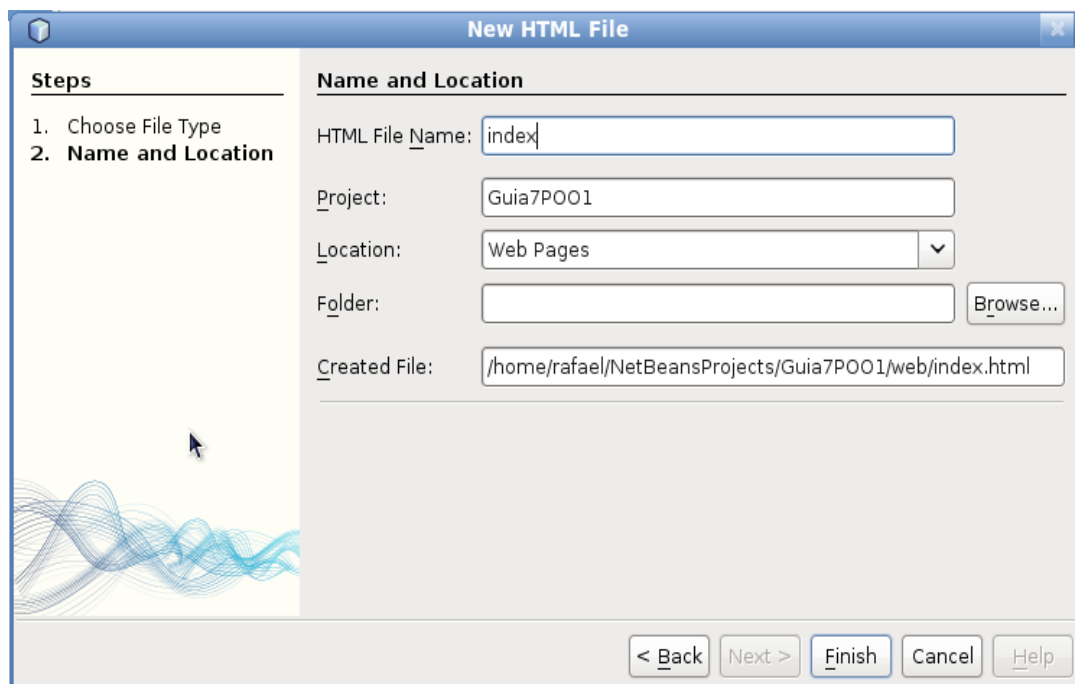
### Paso 3

Eliminación de la pagina index.jsp y creación de la pagina index.html, esto lo hacemos para que el servidor no procese la pagina index.jsp.

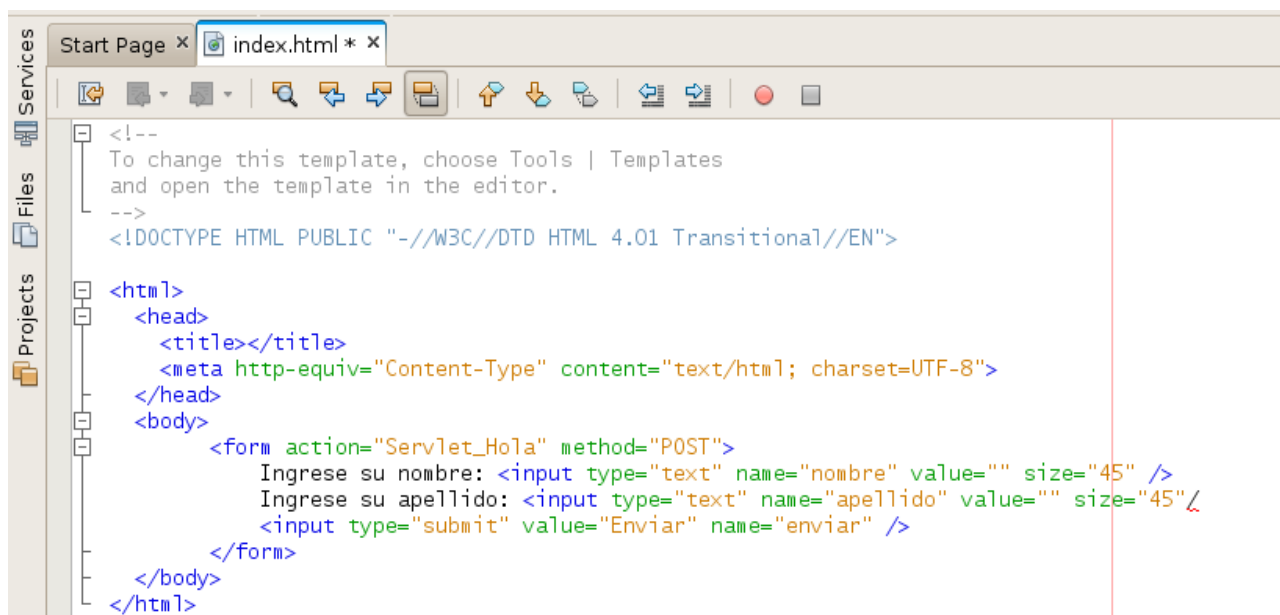
1. Situarce en la pagina y damos click en la tecla suprimir .
2. Ahora crear la nueva pagina JSP, damos click derecho sobre **“Web Pages”**, elegimos la opción **“New”** y seleccionamos **“JSP”** (ver la siguiente imagen).



3. Aparecerá una pantalla como la siguiente en la cual, ingresaremos como nombre de la pagina index, la extensión sera agregada automáticamente.



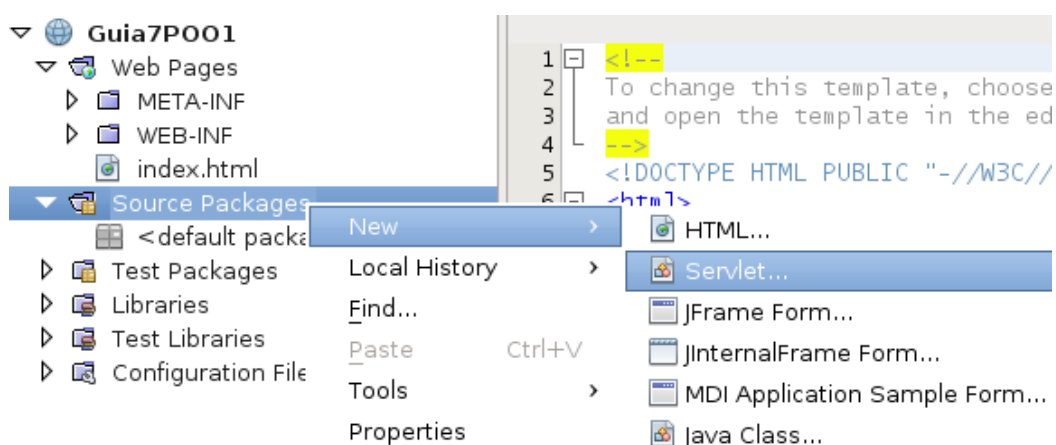
- Ahora modificaremos el código de la pagina index.html de tal manera que quedara de la siguiente manera.



```
<!--
To change this template, choose Tools | Templates
and open the template in the editor.
-->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

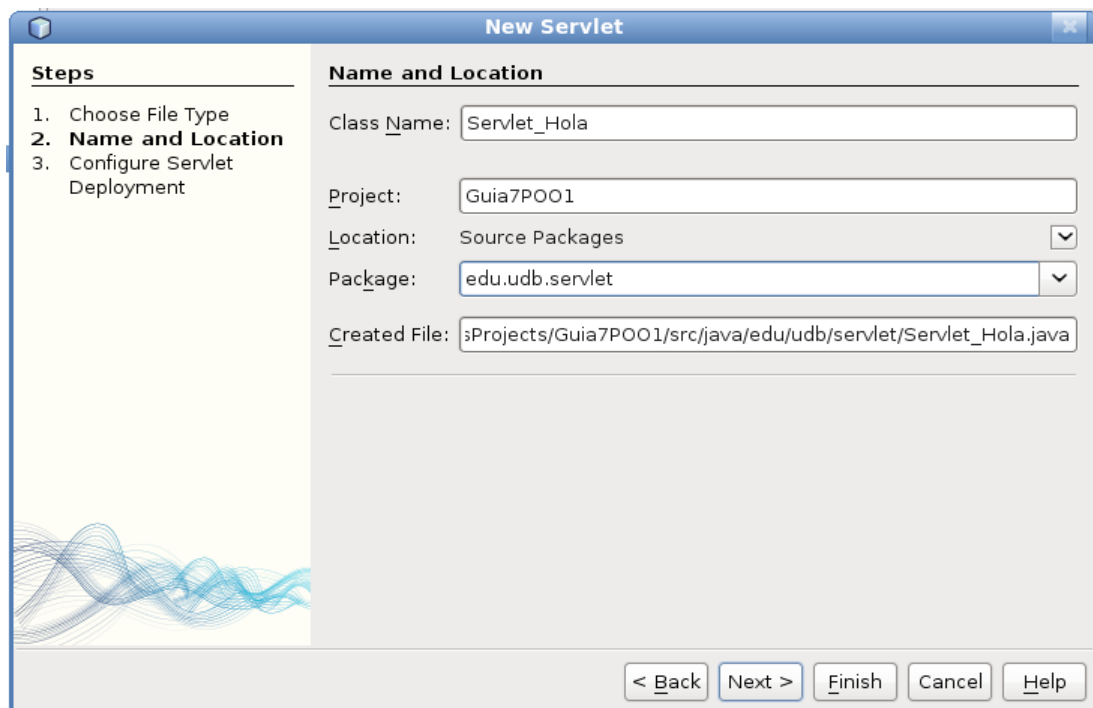
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<form action="Servlet_Hola" method="POST">
  Ingrese su nombre: <input type="text" name="nombre" value="" size="45" />
  Ingrese su apellido: <input type="text" name="apellido" value="" size="45" />
  <input type="submit" value="Enviar" name="enviar" />
</form>
</body>
</html>
```

- Como siguiente paso, se creara el servlet, para ello ,dar click derecho en la carpeta “**Source Packages**”, seleccionamos **New \ Servlet...**



- Se abre un diálogo que nos solicita nombre y paquete del servlet.
  - En nombre, hay que ingresar el mismo nombre del atributo action del formulario creado anteriormente, pues este será el servlet que recibirá los datos enviados por el formulario HTML. En nuestro caso, según indicamos en el form: **Servlet\_Hola**.
  - En paquete se puede ingresar “**edu.udb.servlet**”.

- Dados el nombre del servlet y el paquete, hacemos clic sobre Finish.



Finalizado esto, automáticamente crea una clase con el nombre de servlet dado (Servlet\_Hola para nosotros), que hereda de HttpServlet. Además redefine (override) algunos métodos (doGet, doPost, getServletInfo) y los rellena con un poco de código. Además, crea un método processRequest (invocado desde los métodos doGet y doPost) para procesar los formularios que llegan por los métodos GET y POST.

Nosotros, en este ejemplo, nos limitaremos completar con unas pocas líneas (pues la mayoría la completó automáticamente el NetBeans) el método processRequest para que cree una página HTML que será la respuesta del formulario enviado.

2. Ahora modificaremos un poco el servlet, para ello buscar el método **“processRequest”** y modificamos el código que se encuentra dentro del **“try”**, quedando de la siguiente manera.

```
try {  
  
    out.println("<html>");  
    out.println("<head>");  
    out.println("<title>Servlet Servlet_Hola</title>");  
    out.println("</head>");  
    out.println("<body>");  
    out.println("<h1>Resultado de Servlet_Hola</h1>");  
    out.println("<p>");  
    out.println("<b>Nombre de la persona: </b>" +  
request.getParameter("nombre").toString()+"<br>");  
    out.println("<b>Apellido de la persona:  
</b>"+request.getParameter("apellido").toString());  
    out.println("</p>");  
    out.println("</body>");  
    out.println("</html>");  
  
}
```

#### Paso 4

Ahora se procederá a correr la aplicación.

Para finalizar, vamos a ejecutar el proyecto, podemos hacerlo desde el menú **Run** o haciendo abriendo el menú contextual del proyecto (desde el explorador de proyectos) y seleccionando **Run**.

- Al ejecutar el proyecto se abrirá el browser predeterminado con la página index.html (la que tiene el formulario):
- Si ingresamos nuestro nombre en la caja de texto y damos click en Enviar, el formulario se envía al servlet, quien se ejecuta y nos devuelve una nueva página, con un dato, en particular, cargado dinámicamente, con los valores ingresados en el formulario del index.html.

#### Servlet con JDBC

Para esta parte utilizaremos la base de datos creada en la guía 4.

##### Paso1

Para este punto crearemos una pagina jsp llamada “**ingresaremp.html**”, la cual contendrá el siguiente código.

```
<html>
<head>
<title>Ingresar Empleados</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
  <form action="Servlet_Ingresaremp" method="POST">
    Ingrese su codigo: <input type="text" name="codigo" value="" size="10"><br>
    Ingrese sus nombres: <input type="text" name="nombre" value="" size="45" /><br>
    Ingrese sus apellido: <input type="text" name="apellido" value="" size="45"/><br>
    Ingrese su Telefono: <input type="text" name="telefono" value="" size="10" /><br>
    <input type="submit" value="Enviar" name="enviar" /><br>
  </form>
</body>
</html>
```

##### Paso2

Ahora crea un Servlet en el paquete creado anteriormente (**edu.udb.servlet**) , llamado “**Servlet\_Ingresaremp**”. De este servlet solo tendrá que modificar el método “**processRequest**” y quedara de la siguiente manera:

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();

    ResultSet rs = null;
    Connection conexion = null;
    String ids=request.getParameter("codigo");
    String nombre=request.getParameter("nombre");
    String apellido=request.getParameter("apellido");
    String telefono=request.getParameter("telefono");
```

```

try {
    //Leemos el driver de Mysql
    Class.forName("com.mysql.jdbc.Driver");

    // Se obtiene una conexión con la base de datos.
    conexion = DriverManager.getConnection (
        "jdbc:mysql://localhost/Gui4_LPIII","root", "rafael");

    // Permite ejecutar sentencias SQL sin parámetros
    Statement s = conexion.createStatement();
    s.executeUpdate("Insert into Empleados
values("+ids+",\""+nombre+"\", \""+apellido+"\", \""+telefono+"\)");

    rs = s.executeQuery ("select * from Empleados");
    //Decimos que nos hemos conectado
    out.println("<html>");
    out.println("<body>");
    out.println("<h1>Datos Ingresados Exitosamente</h1>");

    out.println("<table align='center' with='75%' border=1>");
    out.println("<tr><th>Codigo</th><th>Nombres</th><th>Apellidos"+
    "</th><th>Telefono</th></tr>");
    while (rs.next()){
        out.println("<tr><td>"+rs.getInt("Codigo")+ "</td><td>"+
        rs.getString("Nombre")+ "</td><td>"+rs.getString("Apellidos")+ "</td><td>"+
        rs.getString("Telefono")+ "</td></tr>");
    }
    out.println("</table>");
    out.println("</body>");
    out.println("</html>");

    conexion.close();
}
catch (ClassNotFoundException e1) {
    //Error si no puedo leer el driver de Oracle
    out.println("ERROR:No encuentro el driver de la BD: "+
    e1.getMessage());
}
catch (SQLException e2) {
    //Error SQL: login/passwd mal
    out.println("ERROR:Fallo en SQL: "+e2.getMessage());
}
finally {
    out.close();
}
}

```

### Paso 3

1. No se le debe olvidar Agregar el driver de **Mysql** al proyecto y ademas debe de importar la siguiente librería **java.sql.\*** , cada uno de estos pasos fueron vistos en la guía numero cuatro.



- Ahora lo único que falta darle **Run** a la pagina “**ingresaremp.html**”, debe aparecer un formulario el cual debe ser llenado, para luego darle enviar, eso permitirá que el servlet se conecte a la base de datos y pueda insertar los valores digitados en el formulario.

## Manejo de Sesiones

### Paso 1

Crear una nueva pagina html llamada “**login.html**” que contendrá el siguiente código:

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>JSP Page</title>
</head>
<body>
  <h1>Hello World!</h1>
  <form action="GeneraSession" method="POST">
    Ingrese su Usuario: <input type="text" name="usuario" value="" size="45" /><br>
    Ingrese su Password: <input type="password" name="password" value=""
size="45"/><br>
    <input type="submit" value="Enviar" name="enviar" />
  </form>
</body>
</html>
```

Como se puede observa en el **action** debe ir el nombre del servlet que genera la sesión.

### Paso 2

Creamos el Servlet con el nombre “**GeneraSession**”, el paquete sera el mismo utilizado en los puntos anteriores, modificamos el método **processRequest** para que quede de la siguiente manera.

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
        String usuario=request.getParameter("usuario");
        String password=request.getParameter("password");
        if (usuario.equals("rafael") && password.equals("torres")){
            HttpSession session_actual=request.getSession(true);
            session_actual.setAttribute("USER", usuario);
            response.sendRedirect("principal.jsp");
        }
        else{
            response.sendRedirect("login.html");
        }
    } finally {
        out.close();
    }
}
```

En este Servlet se encargada de recoger del usuario y la clave enviados desde el formulario. Una vez recibidos se almacenan en dos variables(“usuario” y “password”) de tipo String. A continuación se comparan con los valores correctos del usuario y la clave.

Si esta comprobación es correcta se crea un objeto de tipo session y se guarda el valor en la variable “usuario” en la sesión mediante el método setAttribute().

A continuación y mediante la opción response.sendRedirect(**“login.html”**), se redirecciona al usuario a la página pasada por parámetro.

### Paso 3

Ahora crearemos la pagina jsp que verificar si la sesión esta activa, la pagina sera llamada **“principal.jsp”** y de be quedar de la siguiente manera.

```
<%  
  
    HttpSession session_actual=request.getSession();  
    String usuario =(String) session_actual.getAttribute("USER");  
    if (usuario==null){  
        response.sendRedirect("login.html");  
    }  
%>  
<html>  
  <head>  
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
    <title>JSP Page</title>  
  </head>  
  <body>  
    <h1>Hello World!</h1>  
    <h2>  
      Bienvenido: <%=usuario%>  
    </h2>  
  </body>  
</html>
```

## III. EJERCICIOS COMPLEMENTARIA

- Crear una aplicación en la cual tome de apoyo el ejemplo de sesiones, solo que ahora el usuario y el password sera leído desde una base de datos llamada “User” y la cual tendrá una tabla con los campos “usuario y password”, de ahí todo el funcionamiento es similar al del ejemplo.
- Utilizando javascrit valide el formulario **“ingresaremp.html”**
- Pasar los ejercicios complementarios de la de JDBC a servlets.